Java • Linux • C/C++ • Win32 • XML • RTOS

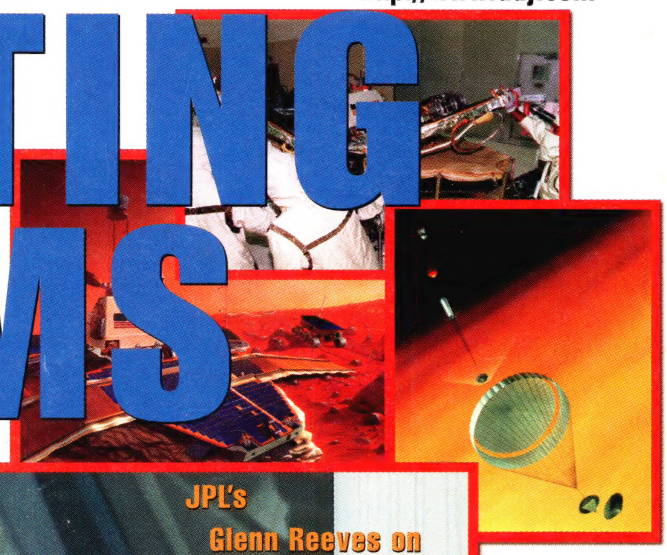
#305 NOVEMBER 1999

Dr. Dobb's JOURNAL

SOFTWARE TOOLS FOR THE PROFESSIONAL PROGRAMMER

http://www.ddj.com

OPERATING SYSTEMS



- Real-Time Linux & IPC
- Extending the PalmOS
- A Ray Tracer for BeOS
- Inside the PowerTV OS
- Real-Time Java

JPL's Glenn Reeves on RTOS & the Mars Pathfinder!

CORBA Meets XML

Machine Learning & Agent-Based Computing

Java & Design by Contract

$4.95US $5.95CAN

11>

0 71486 01051 7

A Miller Freeman Publication

Microsoft
NOW WITH WINDOWS 2000 READINESS KIT
soft
ual
Studio 6.0
Microsoft®
VISUAL®
Studio®
Developer Facts
Serving Size 1 package
Windows 2000 Readiness
Information 100%
Training 100%
Deployment Tools 100%
Tools 100%
Visual Basic 100%
Visual C++ 100%
Visual InterDev 100%
Visual J++ 100%
Visual FoxPro 100%
Total Fat 0%
INGREDIENTS: SCALABILITY, MANAGEABILITY

Everything a growing developer needs for Windows 2000.

Now getting the information, training, and tools you need to get ready for Microsoft® Windows® 2000 is easy. Just pick up Microsoft Visual Studio® 6.0 and your Readiness Kit is in the box. Or if you already have Visual Studio 6.0, you can download a Readiness Kit at **msdn.microsoft.com/vstudio/win2000**

Where do you want to go today?® **Microsoft**

© 1999 Microsoft Corporation. All rights reserved. Microsoft, Visual Basic, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual Studio, the Visual Studio logo, *Where do you want to go today?*, Windows and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

NOTHING BEATS THE FEELING OF

SUCCESS

C++ components from Rogue Wave: the choice of champions!

It can be a stretch to create portable, scalable apps and deliver them on time. And building and testing your own low-level C++ components can really put your game at risk. Why not let Rogue Wave help you cover all the bases? With the integrated and reusable classes in Tools.h++ Professional, Threads.h++, and Standard C++ Library, your development team can get a head start on building a solid, high-performance foundation for every application. Know the score: use C++ components from Rogue Wave, and your whole team will be batting 1000!

Save the day!

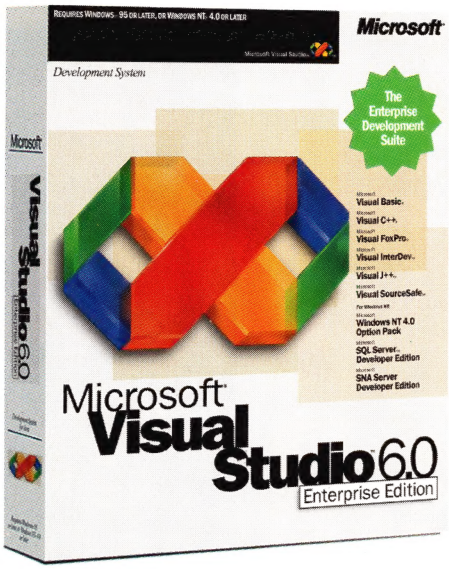
- ✔ Fundamental classes
- ✔ Networking classes
- ✔ Multithreading classes
- ✔ Solutions for Java/C++ interoperability

Get the stats. Download your free demos and white papers now at:
www.roguewave.com/ad/catch

Rogue Wave SOFTWARE
Components Without Limits

Rogue Wave: 800-487-3217 • D.A.CH.: +49-6103-59 34-0 • UK: +44-118-988 0224
France: +33-1-4196 2626 • Italy: +39-02-3809 3288 • Rest of Europe: +31-20-301 26 26

Rogue Wave and .h++ are registered trademarks of Rogue Wave Software, Inc. All other trademarks are the property of their respective holders.



OBJECTIVE STUDIO

User interface tools for the serious enterprise developer.

MFC – ActiveX – Java – UML – ATL

Download your free demos and white papers now at
www.roguewave.com/ad/studio

Rogue Wave SOFTWARE
Components Without Limits

800-924-4223
sales@roguewave.com

Rogue Wave is a registered trademark of Rogue Wave Software, Inc. All products and brand names are trademarks and/or registered trademarks of their respective holders.
Box shot(s) reprinted with permission from Microsoft Corporation.

Dr. Dobb's JOURNAL
SOFTWARE TOOLS FOR THE PROFESSIONAL PROGRAMMER

CONT

FEATURES

A CONVERSATION WITH GLENN REEVES 21
by Jack J. Woehr
Pathfinder was a decade-long program of robotic exploration of the planet Mars. Glenn Reeves, the mission's "Flight Software Cognizant Engineer," chats with Jack Woehr about Pathfinder and what's involved in writing software for extraterrestrial exploration.

LINUX, REAL-TIME LINUX, & IPC 32
by Frederick M. Proctor
When dealing with real-time systems, IPC overhead becomes important. Fred examines two of the best IPC mechanisms available under Linux—FIFO and shared memory.

INSIDE WINDOWS NT SYSTEM DATA 40
by Sven B. Schreiber
Sven untangles the mysterious Windows NT *NtQuerySystemInformation* function.

EXTENDING THE PALMOS 50
by Greg Winton
Packaging code into shared libraries can enhance code reuse and reduce in-memory footprints. Greg takes a look at the Palm shared library model and presents a shared library you can use.

THE POWERTV OPERATING SYSTEM 56
by Morgan Woodson
The PowerTV operating system is designed for set-top boxes and hybrid fiber/cable networks. Morgan provides an overview of the OS, then presents a weather/traffic camera viewer application.

THE BERAYS RAY TRACER 62
by Regan Russell
Regan presents BeRays, an object-oriented ray-tracer application that makes it easy to plug in new rendering ideas.

DESIGNING THE LEGOS MULTITASKING OPERATING SYSTEM 66
by Markus L. Noga
LegOS, the tiny, yet powerful operating system Markus built as an alternative to the RCX firmware, supports features such as preemptive multitasking, dynamic memory management, POSIX semaphores, infrared (IR) communication, and more.

Photographs of Glenn Reeves by Pat Johnson Studios. All JPL/NASA project images courtesy of NASA/JPL/Caltech.

Bill Joy on Openness and Community

Open Source Methodologies at Work

Interviews with Larry Wall, Brewster Kahle, Guy Kawasaki, Tim O'Reilly, Tom Christiansen, *and more...*

For more see
http://www.ddj.com/technetcast/

EMBEDDED SYSTEMS

JBED: JAVA FOR REAL-TIME SYSTEMS 78
by Jörgen Tryggvesson, Torbjörn Mattsson, Hansruedi Heeb
Jbed, a small, fast Java Virtual Machine for embedded real-time systems, includes a complete real-time operating system.

INTERNET PROGRAMMING

MACHINE LEARNING & AGENT-BASED COMPUTING 88
by Zhimin Ding and Li Liu
Zhimin and Li examine the application of machine-learning technology to control software agents, and present MLEngine—a general-purpose AI engine with real-time learning capability.

DR. DOBB'S JOURNAL (ISSN 1044-789X) is published monthly by Miller Freeman, Inc., 600 Harrison Street, San Francisco, CA 94017; 415-905-2200. Periodicals Postage Paid at San Francisco and at additional mailing offices. SUBSCRIPTION: $34.95 for 1 year; $69.90 for 2 years. International orders must be prepaid. Payment may be made via Mastercard, Visa, or American Express; or via U.S. funds drawn on a U.S. bank. Canada and Mexico: $45.00 per year. All other foreign: $70.00 per year. POSTMASTER: Send address changes to *Dr. Dobb's Journal*, P.O. Box 56188, Boulder, CO 80328-6188. GST (Canada) #R124771239. Canada Post International Publications Mail Product (Canadian Distribution) Sales Agreement No. 0548677. FOREIGN NEWSSTAND DISTRIBUTOR: Worldwide Media Service Inc., 30 Montgomery St., Jersey City, NJ 07302; 212-332-7100. Entire contents © 1999 by Miller Freeman, Inc., unless otherwise noted on specific articles. All rights reserved.

ENTS

NOVEMBER 1999
VOLUME 24, ISSUE 11

SOFTWARE CAREERS

TRENDS IN HIGH-TECH HIRING PRACTICES S2
by Reginald B. Charney
Where should you focus your career-development efforts? Reg analyzes technical employment listings to find the answer.

THE WALDEN PROGRAMMER ANALYST APTITUDE TEST S6
by Bruce Winrow
The Programmer Analyst Aptitude Test from Walden Testing has become an essential tool for evaluating high-tech hires.

GAUGING CORPORATE CULTURE S16
by Greg Scileppi
Sometimes "fitting in" is more important than the money you make. Greg shares hints and tips for determining corporate culture before you sign on to a new job.

CAREERS IN SYSTEM ADMINISTRATION S20
by Barbara Dijker
Good system administrators, who possess problem solving skills, people skills, and organizational skills, are in demand in the high-tech job marketplace.

PROGRAMMER'S TOOLCHEST

XML AND CORBA 98
by Dirk Hamstra
The XML|IT toolkit from CareFlow lets you automatically tag results returned from calls to CORBA services, then format them using XML.

COLUMNS

PROGRAMMING PARADIGMS 103
by Michael Swaine
According to Arthur C. Clarke, any sufficiently advanced technology looks like magic. Michael looks at some of the magical moments that are shaping the world.

C PROGRAMMING 107
by Al Stevens
Al has a thing or two to say about book publishers and programming books, then he turns his attention to this month's project—TYFC, a framework to "Teach Yourself Framework Classes."

JAVA Q&A 113
by Krishnan Rangaraajan
Does Java support Design by Contract? Not directly, but as Krishnan shows here, support is still possible.

ALGORITHM ALLEY 119
by Tim Kientzle
Unsharp masking is a photographic technique that increases the sharpness of photographic images. Tim presents an algorithm that implements this concept.

DR. ECCO'S OMNIHEURIST CORNER 123
by Dennis E. Shasha
One man's Calabaza is another man's pocket watch, as Ecco and Liane discover this month.

PROGRAMMER'S BOOKSHELF 127
by Gregory V. Wilson
Greg examines *Mr. Bunny's Big Cup o' Java, The Inmates are Running the Asylum, Ready-to-Run Java 3D, The Java 3D API Specification, Advanced CORBA Programming with C++*, and *Inside the JavaOS Operating System.*

FORUM

EDITORIAL 8
by Jonathan Erickson

LETTERS 10
by you

NEWS & VIEWS 18
by Nick Baran

OF INTEREST 132
by Nick Baran and Amy Lincicum

SWAINE'S FLAMES 144
by Michael Swaine

RESOURCE CENTER

As a service to our readers, source code and related files, and author guidelines are available at http://www.ddj.com/. Source code is also available via anonymous FTP from ftp.ddj.com (199.125.85.76). Letters to the editor, article proposals/submissions, and inquiries can be sent to editors@ddj.com, faxed to 650-358-9749, or mailed to *Dr. Dobb's Journal*, 411 Borel Ave., Suite 100, San Mateo, CA 94402-3522.

For subscription questions, change of address, and orders, call 800-456-1215 (U.S. or Canada). For all other countries, call 303-678-8475 or fax 303-661-1181. E-mail subscription questions to ddj@neodata.com or write to *Dr. Dobb's Journal*, P.O. Box 56188, Boulder, CO 80322-6188.

Back issues may be purchased for $9.00 per copy (includes shipping and handling). For issue availability, send e-mail to orders@mfi.com, fax to 785-841-2624, or call 800-444-4881 (U.S. and Canada) or 785-838-7500 (all other countries). Back issue orders must be prepaid. Please send payment to *Dr. Dobb's Journal*, 1601 W. 23rd Street, Suite 200, Lawrence, KS 66046-2700.

Individual back articles may be purchased electronically at http://www.ddj.com/ as ZIP archives.

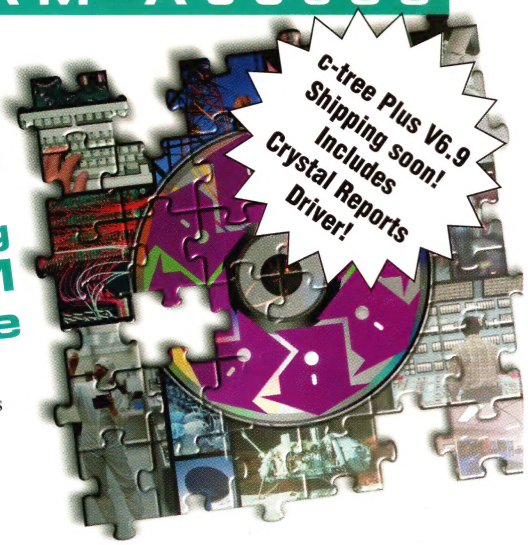
NEXT MONTH

We've got a lock on December when we examine issues involved in computer security.

Fast ISAM Access

Puzzled by database performance issues?

Nothing beats... ISAM performance



Real-world data management solutions are typically more complex when one examines the pieces, than initially recognized by the majority of database programers. All software projects are complex puzzles comprised of many details, most of which are data-related. Often today's "DBMS" solutions sacrifice the speed or control essential for a competitive application.

c-tree Plus®, by FairCom, has been the choice of commercial developers for twenty years precisely because it offers the flexibility and control at the detail level to fit a wide variety of data management needs. Proven on large Unix servers and workstations, c-tree Plus's small footprint and exceptional performance have also made it the engine of choice for professional developers on Windows and Mac. c-tree Plus offers sophisticated ISAM level control with which the developer may define precise data management solutions, making it a perfect fit for any development project requiring specific data handling features.

c-tree Plus® offers the most mature ISAM solution today...

FairCom's c-tree Plus database engine:

- Advanced Indexing Technology
- Complete Source Code
- Complete Transaction Processing
- ODBC Interface
- Over 25 Developer's Servers Included
- Portable Multi-threaded API
- Royalty Free
- Standalone, Multi-user or Client/Server Models
- Thread-Safe Libraries
- Y2K Compliant

The FairCom Server:

A solid, high performance database server that is scalable, portable and offers unequalled control. FairCom has been providing database solutions to the commercial development community for twenty years. You won't find a better solution, with these features and performance anywhere else!

- Client Side Source Code
- File Encryption
- File Mirroring Logic
- Full Conditional Index Support
- Full Heterogeneous Networking
- Multiple Communication Protocols
- Online Data Backup
- Small Memory Footprint
- Flexible OEM Licensing Options
- Source Code Availability

All these platforms supported in one package:

MIPS ABI, DEC Alpha, Sun SPARC, Windows 9X, SCO, 88OPEN, AIX, RS/6000, HP9000, Sun OS, Interactive Unix, Linux (Alpha…), AT&T System V, QNX, Free BSD, OS/2, Mac, Windows NT, Windows 3.1, DOS, Netware NLM, & Banyan VINES.

Don't wait, see for yourself!
USA. 800.234.8180
www.faircom.com
Email: info@faircom.com

FairCom®

Database SOLUTIONS Since 1979

Phone: USA 573.445.6833 - EUROPE +39.035.773.464 - JAPAN +81.59.229.7504 - BRAZIL +55.11.3872.9802

Other company, product and operating platform names are registered trademarks or trademarks of their respective owners.

Dr. Dobb's JOURNAL

SOFTWARE TOOLS FOR THE PROFESSIONAL PROGRAMMER

PUBLISHER
Timothy Trickett

EDITORIAL

EDITOR-IN-CHIEF
Jonathan Erickson
MANAGING EDITOR
Deirdre Blake
MANAGING EDITOR, DIGITAL MEDIA
Kevin Carlson
EXECUTIVE EDITOR
Tim Kientzle
SENIOR EDITOR
Nick Baran
SENIOR PRODUCTION EDITOR
Monica E. Berg
EDITORIAL ASSISTANT
Amy Lincicum
ART DIRECTOR
Margaret A. Anderson
INTERNET BROADCAST PRODUCER
Philippe Lourier
CONTRIBUTING EDITORS
Al Stevens, Tom Genereaux, David Betz, Bruce Schneier, Mark Russinovich, Bryce Cogswell, Ray Duncan, Jack Woehr, Jon Bentley, Dennis Shasha
EDITOR-AT-LARGE
Michael Swaine
PRODUCTION MANAGER
Denise Denis

CIRCULATION

DIRECTOR OF CIRCULATION
Jerry Okabe
GROUP CIRCULATION MANAGER
Michael Poplardo

MARKETING/ADVERTISING

ASSOCIATE PUBLISHER, EAST
Brenner Fuller
SALES DIRECTOR, WEST
Stan Barnes
MARKETING DIRECTOR
Holly Vessichelli
DIGITAL MEDIA MANAGER
Michael Calderon
MARKETING COORDINATOR
Marquita Tinio
ACCOUNT MANAGERS see page 129
Gabriel Rogol, Michael Beasley, Tom Hudner, Michael Kelleher, Linda Guyette, Elizabeth Doherty, Randy Byers, Ron Cordek, Phil Marshall
GRAPHIC DESIGNER
Carey Perez

DR. DOBB'S JOURNAL
411 Borel Avenue, San Mateo, CA 94402-3522
650-358-9500. http://www.ddj.com/

MILLER FREEMAN INC.

CEO/MILLER FREEMAN GLOBAL
Tony Tillin
CHAIRMAN/MILLER FREEMAN INC.
Marshall W. Freeman
PRESIDENT
Donald A. Pazour
EXECUTIVE VICE PRESIDENTS
Darrell Denny, Galen A. Poss, Regina Starr Ridley
SENIOR VICE PRESIDENTS
Wini D. Ragus, Peter Hutchinson
VICE PRESIDENT/PRODUCTION
Andrew A. Mickus
VICE PRESIDENT/CIRCULATION
Jerry Okabe
VICE PRESIDENT/GROUP PUBLISHER
Peter Westerman

Estimated print run 210,000

Miller Freeman
A United News & Media publication

ABP American Buisness Press

Printed in the USA

BPA Consumer Audit

THERE IS A BONAFIDE INTERNET CONSPIRACY TO REVEAL THE SECRETS OF INFORMATION TECHNOLOGY TO THE WORLD. INFORM IT.
YOUR BRAIN IS HUNGRY. InformIT
Devious, diabolical, fascinating and brilliant minds have conspired to make public, all answers to all questions of Information Technology. There is a site growing at blinding speed, designed to burn away the murky veil of doubt that troubles the programmer, databaser, webmeister, networker or configurationist. They know LINUX. They know JAVA. They know HTML. They know who killed JFK. www.informit.com
©1999 informIT. All rights reserved. InformIT is a trademark of Macmillan USA.

"I Said Turn RIGHT, Fool!"

It was with great relief that we safely navigated this summer's end-of-week (EOW) rollover minefield. Dismissed by some as "a Y2K preseason game," EOW rollover was billed by others as a Y2K warm-up that threatened the accuracy of global positioning systems (GPS) around the world. In a nutshell, the EOW rollover problem involved GPS receivers—not the satellites or transmitters—that might have thought the date was January 6, 1980, as GPS time rolled over from August 21 to August 20, 1999.

According to the U.S. Coast Guard Navigation Center (http://www.naven.uscg.mil/gps/geninfo/y2k/), the timescale origin of GPS time started at midnight January 5-6, 1980 (or Julian Day 2,444,244.500). Since a GPS cycle is 1024 weeks (or 7168 days), our first GPS rollover was due to occur at Julian Day (2444244.5+7168)=2,451,412.5—00:00:00 on August 22, 1999, in other words. Putting this another way, that's 7168/365.25=19.6249 years, or 1980+19.625=1999.638. No matter how you count it, it still comes out to August 22, 1999. (For details, see Richard Langley's excellent description of the EOW phenomenon is "The GPS End-of-Week Rollover" at http://www.gpsworld.com/0299/1198innovation.html.)

The "old" Hertz NeverLost GPS.

The upshot of all this was the potential for calamity, since millions of GPS receivers are being used by business, military, and consumers for everything from navigation and telecommunications, to finance and network computing. Of course, there shouldn't have been any problem for GPS receivers with updated Y2K/EOW-rollover-compliant firmware.

EOW rollover panic wasn't rampant. No, unlike Y2K, there were no conferences, books, consultants, newsletters, web sites, or CNN talking heads trying to scare you out of a few bucks. Yes, there were scattered reports of frantic bass fishermen who depend on GPS to pinpoint secret fishing holes. As it turned out, there was little need to worry, at least in the U.S. where virtually no rollover-related problems were reported. That wasn't the case in Japan, however, where hundreds of people were confused and inconvenienced, as screens on car-navigation systems went blank or were filled with random characters. At least we know where all the nonEOW-compliant receivers were. In short, GPS EOW rollover came and went rather quietly—this time around.

I suppose you'd call me a Jonny-come-lately when it comes to this GPS stuff. Gee, even my sister, who is an archeologist, is way ahead of me. Faith has been using GPS for years to identify and locate historical sites in the Negev Desert. (She reports via e-mail, by the way, that she didn't experience any problems with EOW rollover.) In fact, she's on to her second- or third-generation GPS receiver, and insists that they are as necessary to her work as, well, a shovel and a pith helmet.

The extent of my GPS experience, on the other hand, is with Hertz rental cars that have NeverLost In-Car Navigation Systems installed. NeverLost (http://www.hertz.com/serv/us/prod_lost.html), which uses GPS, dead-reckoning, and map-matching technology, is a private-label version of Magellan's proprietary 750NAV vehicle-navigation and driver-information system (http://www.magellangps.com/). More than 50,000 Hertz cars currently come with Magellan hardware and Navigation Technologies (http://www.navtech.com/) databases covering more than 50 U.S. cities.

The Ford Taurus I was driving, for instance, had the entire database for California online. Using a clunky interface, you pressed the cursor pad to select the destination city from a long list of cities, then the street, and finally the address you want to drive to. The system takes a few seconds to compute the best route, then displays on-screen icons and issues audible voice commands to get you going in the right direction. As you approach a required turn, the system alerts you with a beep about a block prior to the actual junction, followed by a voice announcement, such as "Right Turn Ahead." If you miss the turn, you need only press Enter and the system computes a new route for you (and, mercifully, doesn't add the word "...fool.") The Hertz web site and TV commercials are now hawking NeverLost units with a new cursor pad and improved interface—specifically, a QuickSpell feature that lets you go more directly to a city or street by selecting the first letter of the name, rather than scrolling down a long list. However, the Hertz cars I've recently rented still have units with the old interface.

Okay, I admit that until I actually used one, I thought systems like Hertz's NeverLost GPS were nothing more than marketing gimmicks. But they truly are a valuable application of technology that just happens to be fun. Even with the old NeverLost interface, in-car navigation is relatively easy to use, and certainly much more convenient (and safer) than pulling over to the curb and spreading out a map. And as for the EOW rollover interfering with GPS, don't worry—it won't come again for another 19 years.

Jonathan Erickson
editor-in-chief
jerickson@ddj.com

Building setups that include third party technologies is a constant battle.

InstallShield Professional 2000 is the peace plan.

INSTALLSHIELD OBJECTS
make it dramatically simpler to include and deploy third-party technologies.

DEPENDENCY MANAGER
automatically scans your app to identify required DLL's and offers choices on whether (and where) they should be incorporated.

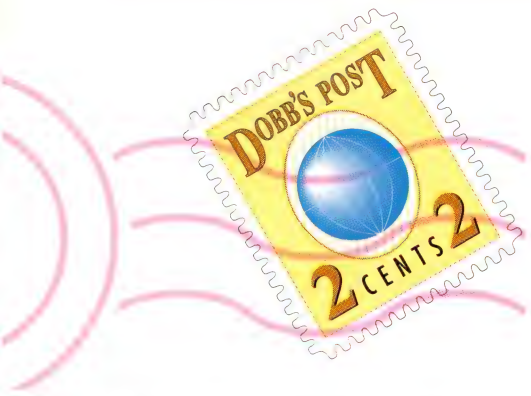
COMPONENT INSTALL/UNINSTALL
deliver individual features, applications or suites that can be easily changed, removed, or reinstalled by the end user. No need for custom DLLs for Uninstallation.

EVENT-BASED SCRIPTING
improved script management structure reduces the amount of code necessary to create setups, so you have more time to customize them.

The code war is over. The newest version of the industry standard for installation development is here: InstallShield Professional 2000 – delivering breakthrough usability and productivity for developers who create setups of any size or complexity, **for any Windows platform** – including Windows 2000. As a developer, your most precious commodity is time. And you'll save plenty of it with the powerful new features in InstallShield Professional 2000; like InstallShield Objects, which take the grunt work out of building installations by making it easy to support established third-party technologies. So don't waste another minute fighting the installation war. Visit **www.installshield.com/peace** or call **800-969-7724** to learn more about InstallShield Professional 2000, download a trial version, or purchase it immediately, and start using the newest version of the industry standard in setup authoring today.

InstallShield®
SOFTWARE CORPORATION

© 1999 InstallShield Software Corporation. All rights reserved. InstallShield is a registered trademark and service mark of InstallShield Corporation. Windows is a registered trademark of Microsoft Corporation. All other trademarks are property of their respective owners. Part No. 211-20035 0999



Programmer Aptitude Testing

Dear *DDJ*,

Readers interested in the "Wolfe programming aptitude test" (see "Letters" *DDJ*, September 1999) might find the history of the test interesting as well. Our Aptitude Assessment Battery Programming test was developed by the late Dr. Jack M. Wolfe, Professor Emeritus of Computer Science at City University of New York. Dr. Wolfe was one of the pioneers of computer science in the 1960s and developed his test to replace the old IBM test which he thought had major deficiencies. The test is a simulation of what a programmer is required to do. It is virtually always accurate in predicting success as a programmer. It measures not only logical ability and problem solving ability, but such qualitative factors as documentation and annotation ability, whether a person is better suited for maintenance or creative programming, whether and how well the candidate desk checks his work and whether he is a faster or slower than average performer. It is available in several languages, in Braille, and in a left-handed version (no kidding). The scoring process took Dr. Wolfe over seven years to perfect and validate. The CIA used it for over 15 years to assess its programmer applicants and validated it; so have thousands of other organizations.

Stephen Silver
Walden Personnel Testing & Training
ssilver@waldentesting.com

DDJ responds: DDJ readers attending the Software Development '99 East Conference (http://www.sdexpo.com/) in Washington, DC (November 10–11) can take the Walden Programmer Analyst Aptitude Test at no charge in a special program sponsored by *DDJ*. If you're interested and are planning on attending the conference, sign up by sending e-mail to editors@ddj.com.

Cross-Platform Design

Dear *DDJ*,

I read with interest Bob Krause's "Cross-Platform Design Strategies" (*DDJ*, June 1999). The issue is not as simple as Bob makes it. While his goals are worthy, the thread classes he describes actually fall into some of the problems he is trying to avoid. The problem is that, in general, OS differences just cannot be cleanly abstracted without losing functionality.

In the thread case, right at the start there is the issue of preemptive thread switching. His thread classes choose to explicitly disable preemptive switching. This is a lowest common denominator decision. That is not to say that it was a bad decision.

The other issue was the block call. By not fully supporting the ability to block with a timeout the abstraction is lost. Either the programmer has to make do without that functionality or has to make OS-specific code to handle the differences.

These abstractions are still the way to achieve a cross-platform application but they are not truly generic since they include assumptions about the functionality that is required and these should be made explicit.

Tim Bond
tbond@phios.com

A Hard Day's Night

Dear *DDJ*,

In "News & Views" (*DDJ*, September 1999), we were told that Pitney Bowes reckons that the average U.S. worker receives 200 e-mails each day and is interrupted by fax, phone, or e-mails about every 10 minutes. I don't think so! A little simple arithmetic — no computer needed — shows that even if the average U.S. worker works 24 hours each day and receives 200 e-mails, then the average interval is 7.2 minutes — just for e-mails. Even programmers don't work 24 hours every day and I doubt that the average worker receives 200 e-mails each day. Perhaps we should be a bit skeptical of reports like these from Pitney Bowes? They aren't perhaps, trying to sell something, are they?

Mike Pidd
M.Pidd@lancaster.ac.uk

Electronic Music

Dear *DDJ*,

When I was about 17 years old, circa 1961, I cobbled together a bunch of contraptions for the French department language lab that sound a lot like the machines Al Stevens described in his "C Programming" column (*DDJ*, September 1999)— raw materials were some Phillips reel-to-reel tape recorders, 117-VAC relays, and switch boxes. Students played a reel on the machine in front of them, another hidden machine with a 20 second (fixed) endless loop recorded the output on one track. Students could turn a rotary switch to stop the playback and hear the loop, another position to record their own rendition on a different track of the loop, flip back and forth until they were happy with it, and go back to the main machine. Not bad for the time.

The department head published this nonsense in, I think, "The Journal of Catholic Education," about 1962-63. If, as it seems, nameless_agency.gov had priority, we will have to print a retraction. Please advise.

Dugald A. Taylor, M.D.
Wichita, Kansas

So What's Wrong with COBOL?

Dear *DDJ*,

Have you also noticed that the simple things we did in Cobol (or Fortran, or BASICA, or whatever) have become next to impossible today? I just want to access my COM port with VB6.0 Learning Edition or update a small text file on a server (without needing Cold Fusion), or stick a few debugging "beeps" into Visual C++. I put a clever little conversational CGI program on my ISP's server to do a calculation (like good old CICS). It was simple and well behaved on my PC, but because I compiled it with a 16-bit compiler, it brought the entire server crashing down. My ISP thinks I'm an idiot. (I'm not.) Turns out Microsoft has a bug in its IIS server that can't deal with an older MS exe file. I have a lot of respect for mainframe hardware and software. You never get a hardware error and when you get a software error, there is a clear cause that you can determine and correct on your own, most of the time. And you can't crash a mainframe (by yourself). They only reboot (called IPL) for a damn good reason. Yes, Cobol is wordy and relatively unsophisticated, but the beauty is that no time was ever lost getting some simple customer file to print or a badly needed summary total flashed on a screen. It never seemed that important in a business setting to run a movie or marquee some stupid message like an old teletype. Is this part of the "dumbing down" trend?

Chris Clement
micrometer@compuserve.com

Dear *DDJ*,

Could you pass along to Philip Hendrickson that I do program in Cobol and C and that I'm not working on Y2K issues, that our major application is written in Cobol — that's right Cobol — and doctors and hospitals use our application everyday and will continue to use our application because it is reliable and runs on a reliable operating system "AIX" by IBM. So since I write small utilities in C and major applications in Cobol I guess that makes me a real programmer who likes Cobol? Oh by the way we use

More about the copy constructor

by Adam Kolawa

This month we will illustrate the dangers of not writing a copy constructor for your code.

Suppose we have a class that keeps employee records, and the pointer member **name** represents the last name of an employee.

```
class Employee {
private:
   char *name;
public:
   Employee(const char *lname = 0);
}
```

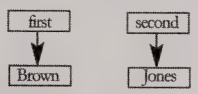
Let's define the constructor for the **Employee** class:

```
Employee::Employee(const char *lname)
{
   if(name) {
      name = new char[strlen(lname) + 1];
      strcpy(name, lname);
   }
   else name = 0;
```

Let's assume we have created two employees "first" and "second", with last names Brown and Jones:

```
Employee first("Brown");
Employee second("Jones");
```

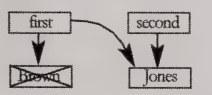
The layout of the data in the memory will be as displayed below. The first employee record has memory attached that contains the name Brown, and the second has memory attached that contains the name Jones:



Let's assume that for some reason we're assigning the second record to the first employee.

```
first=second
```

The second record will point to Jones, but the first will also point to Jones because the default copy constructor copies memberwise assignments:



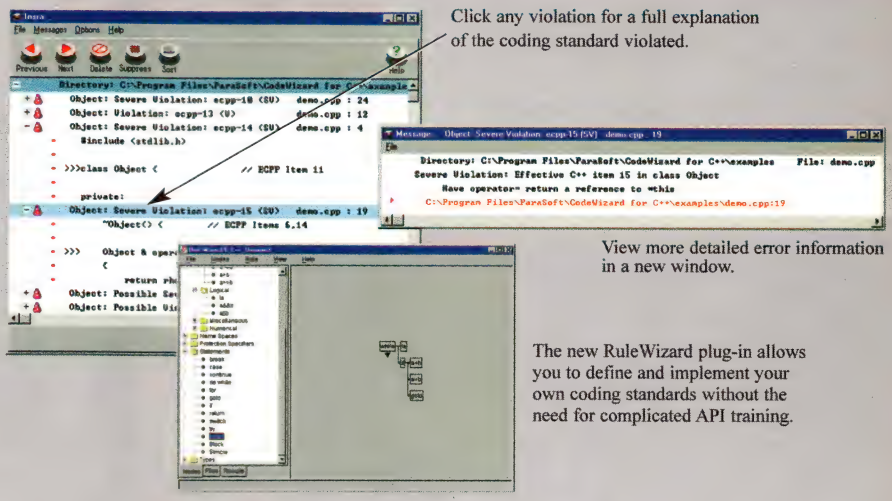
Now let's assume that for some reason, we want to change the first letter of Jones to a "K" to make the name "Kones":

```
first.name[0] = 'K';
```

The record will change for the first and the second employee. This is probably not what we intended. Fortunately, you can avoid this problem by writing a copy constructor, which will create new memory and copy to it.

Next month we will illustrate another dangerous coding construct. Thank you for reading.

Adam Kolawa is President and CEO of the ParaSoft Corporation.
He can be reached at ukola@parasoft.com



Would you believe that CodeWizard® can prevent errors in your code and cut your debugging by 25%?

*"CodeWizard is the code reviewer without the attitude; it knows all the rules." -- **Suzette LaGray** Siemens ElectroCom L.P.*

CodeWizard® is a tool that helps you prevent errors by automatically performing static analysis on C++ code. The first thing you need to know about CodeWizard is that it can help you do less of the thing you hate the most: debugging. With CodeWizard, preventing errors before they happen is painless.

CodeWizard automatically enforces over 70 C++ coding standards, which are language-specific rules that help you to avoid dangerous coding constructs. The standards are borrowed from the writings of Scott Meyers and other industry experts. CodeWizard clearly displays each of your violations and gives you all the information you need to fix them, including suggestions on better coding constructs. No more wondering about weaknesses in your code. Follow CodeWizard's coding suggestions, and you'll immediately begin to write cleaner, more reliable code.

The key to CodeWizard is that you can configure the program to avoid overworking yourself. With a few clicks, you can tell CodeWizard to enforce only the standards that are most relevant to the current project. Using the new RuleWizard™ plug-in, you can even write your own coding standards with point-and-click convenience. Use CodeWizard and RuleWizard together, and you'll have a personalized tool for your entire development team. This is a tool that conforms to you–not the other way around.

You won't have any problem adding CodeWizard to your arsenal of development tools, because it installs directly into Windows® and UNIX development environments. In Microsoft® Developer Studio®, you'll begin testing with a single click of the CodeWizard icon. On UNIX platforms, CodeWizard is a wrapper around the compiler. Learn a few simple commands, and you'll be on your way to cleaner code.

If you would like to prevent errors, reduce debugging, and go home on time for a change, download a copy of CodeWizard today at www.parasoft.com/believe.htm, or call (888)305-0041 for more information.

ParaSoft®

www.parasoft.com/believe.htm

ParaSoft is a registered trademark, CodeWizard is a registered trademark of ParaSoft Corporation, and RuleWizard is a trademark of ParaSoft Corporation. Microsoft Developer Studio and Windows are registered trademarks of Microsoft in the US and other countries.

(continued from page 10)
MicroFocus Cobol for our billing software and RM Cobol for our charting software.

Richard Shirey
gizzmaxx@cobweb.net

Dear *DDJ*,
Everyone has been saying "Cobol is dead" for years now. I work for the 32nd largest credit card processing bank. It is a private company. There are only 250 of us, including the HR and receptionists. But I make good money and am happy. I am over 60— programming (in Cobol on Tandem— oops Compaq) for over 15 years. My degree is in business.

In addition to the processing switch (millions of transactions per month), we do license work for large companies whose names you would know, refunds for a home improvement, multiple applications for a Texas-based company. All they have in common is large, complex systems that work well, run on boxes that are 24/7 transaction processors with fault tolerance required— and they are written in Cobol. We also use PC workstations, telecommuting, NT servers, Internet mail, and all the other "good" stuff that real programmers write. But they are tools.

Even a nonprogrammer can support self-documenting code simply by reading the source. Then when a new enhancement is required, or an "undocumented feature" is discovered (read some real-time employee tried to do something we didn't think of in QA and got a weird response) a programmer makes the changes.

All you "real" programmers keep trucking. I enjoy my sailboat, have a wife, kids, grandkids, two dogs, and a cat that thinks he's a dog. While you are stressed with three cell phones, two beepers, a laptop, and who knows what else, I will continue to code in Cobol until I am 90, telecomputing from my bed in the nursing home.

GeorgeVol@aol.com

Porting to CE

Dear *DDJ*,
After reading the article "Porting Communications Software to Windows CE", by Oliver Deiner (*DDJ*, July 1999), I have some feedback and solutions to a few of the points he raised.

Regarding overlapped I/O support in CE: It is true that the feature is not supported directly by the file I/O functions (*CreateFile*, *ReadFile*, and so on), but it is indirectly supported for serial communications at the device driver level. The Microsoft Knowledge Base article Q175551 "INFO: Serial Communications on Windows CE HPC devices" discusses the exact details of the support provided.

Support for TAPI inbound connections has been the big complaint about CE 2.x and it has been publicly stated that the feature will definitely be available in the next major release of CE. See Doug Boling's article "Updated with New Kernel Features, Windows CE 3.0 Really Packs a Punch" (*Microsoft Systems Journal*, July 1999). I wrote major portions of the Visual C++ for Windows CE debugger so his comments with regard to remote communication are near and dear to my heart. Oliver's criticisms are completely valid, but the easy solution is to use an Ethernet connection instead of the serial port. There is a hack to get this working in VCCE 2.0 and it is fully supported with Platform Manager in the current version, VCCE 6.0. The slides and video from my 1999 CE Developer's Conference presentation "Tips and Tricks for Debugging with VCCE" are available at http://events. microsoft.com/isapi/events/pages.asp?s=63814&p=623 and give details on how to set this up. Thanks for an interesting and informative article on Oliver's experience. It's feedback such as this that will help improve future versions of both the OS and the development environment.

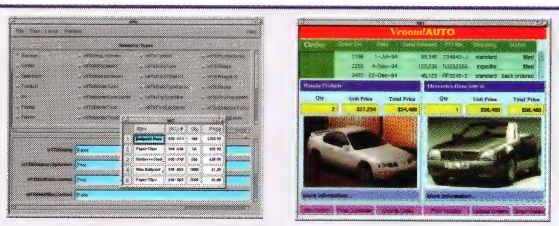
Andrew Tucker
andrewt@bsquare.com

DDJ

❑ Build your own
❑ Shareware
☑ XRT Widgets for Motif

Professionals know there's only one answer to Motif interface development: XRT Widgets from KL Group.

XRT/table – Multi-purpose lists, tables and forms



- Data may be supplied programmatically, read from a file, entered by users, or supplied at run-time using a callback
- Supports table sizes of up to 2 billion rows by 2 billion columns
- Advanced Widget-in-Cell technology lets you efficiently manage widgets and pixmaps
- Includes label widget that supports rotated text and PostScript output
- Fast preview scrolling
- Royalty-free

XRT PDS also includes:
XRT/graph – Wide variety of 2D charts and graphs
XRT/3d – Sophisticated 3D charts and graphs
XRT/gear – Essential add-on widgets and utilities
XRT/field – Effortless data-entry and validation

XRT supports most major UNIX platforms, VMS, and Windows via Interix and NuTCRACKER. Integrates with major GUI builders including UIM/X, Builder Xcessory, TeleUSE, View.h++, X-Designer and XRunner.

FREE XRT Widget Tips and Progress Bar at:
www.klgroup.com/widgets

KL GROUP
Software Development Productivity

KL Group Inc.
1-888-361-9161
Tel: 416-594-1026
Fax: 416-594-1919

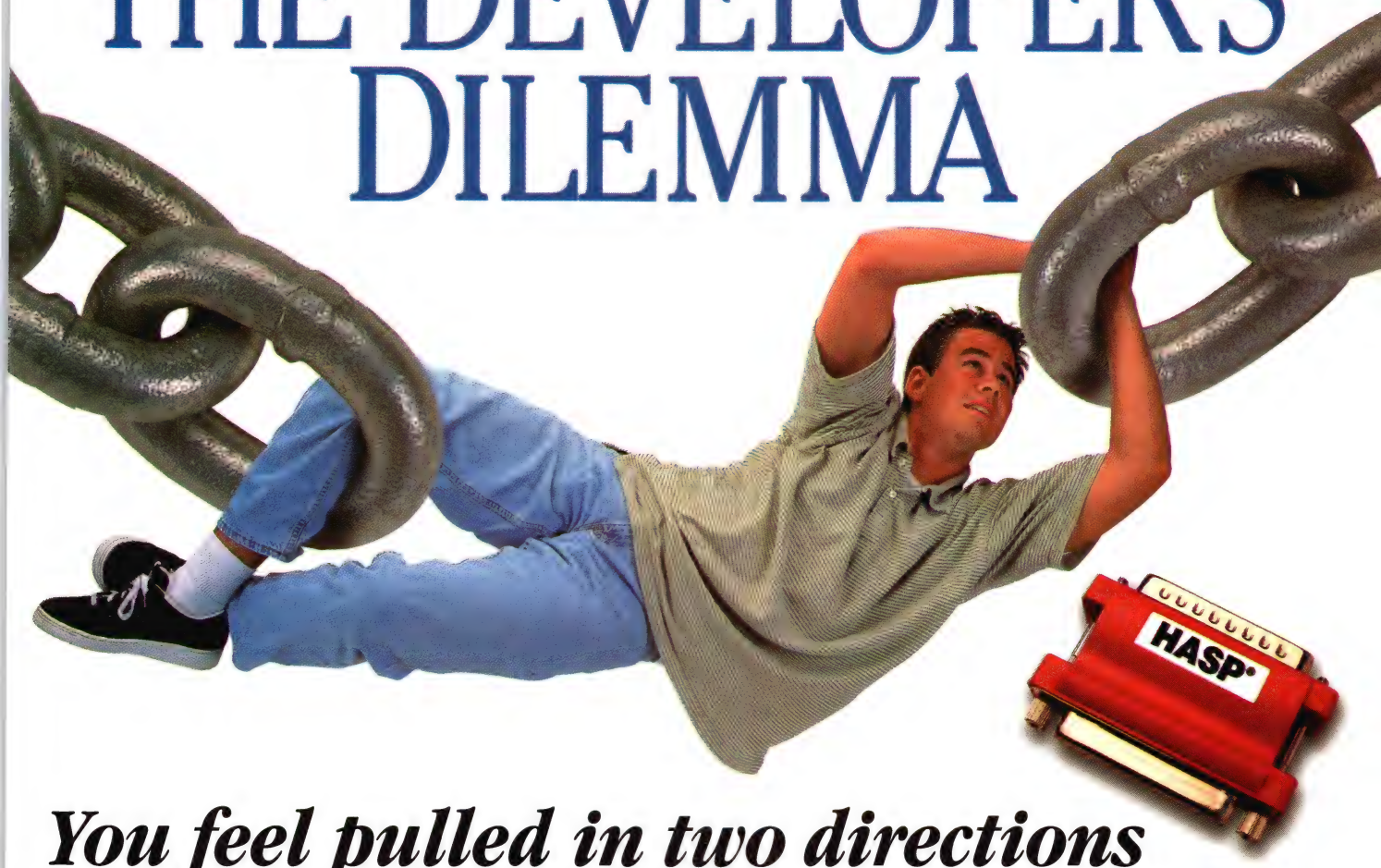
KL Group Europe
Tel: +31 (0)20 679 95 03
Fax: +31 (0)20 470 03 26

1997 Editor's Award UNIX

XRT, XRT PDS, XRT/gear, XRT/graph, XRT/field, XRT/table, XRT/3d and XRT/graph are trademarks of KL Group Inc. All other products are trademarks or registered trademarks of their respective companies. DDDD

Protect or Not Protect?

THE DEVELOPER'S DILEMMA



You feel pulled in two directions

On one hand, thanks to piracy and unauthorized use, you know you're losing one dollar for every dollar you make.

On the other hand, you're afraid you'll lose customers if you try to protect your applications.

HASP® absolutely destroys that dilemma!

HASP gives you outstanding protection, plus it's extremely friendly to your end-users.

In fact, National Software Testing Laboratories compared HASP to our major competitor and declared HASP Number One.*

Twenty-five thousand developers have solved their dilemma by choosing HASP. Why don't you?

Call for your Developer's Kit today!

www.eAladdin.com

ALADDIN®

Securing The Global Village

USA 1-(800) 223-4277
Intl. 972-3-636-2222

Offices in: United States, Israel, Germany, United Kingdom, Netherlands, Japan and South Africa. Fifty-plus distributors serving more than 100 countries.

HASP® is a registered trademark of Aladdin Knowledge Systems, Ltd. Copyright 1999 • *Full copy of study available upon request.

Subscribe Today... at www.pparadise.com

Island Insider™
Weekly E-mail Newsletter

Weekly Special Offers & Announcements

Paradise Picks

FarPoint's Spread 3
by FarPoint Technologies

Use FarPoint's Spread 3 to create powerful database front-ends, manage the display and entry of two billion items using the enhanced Virtual Mode, print reports using Print Preview, perform calculations, import/export Excel 97 files (32-bit versions), export HTML files (32-bit versions), sort data, support OLE Drag and Drop, or take advantage of its unparalleled cell-level formatting, including twelve built-in cell types. Unmatched power, flexibility, and speed combine to make Spread the perfect component for any application.

Paradise No. F02 0120-GL

$265.95

Wasp Bar Code ActiveX & DLL
by Informatics

- Supports all major bar code symbologies. Optional support for new 2D symbols (e.g. PDF417, MaxiCode, DataMatrix)
- Render spec compliant bar codes to any device context
- Two ActiveX controls fully support Drawing and Font DLLs and can be bound to a database.

Exclusive Offer. FREE Wasp Bar Code Scanner ($175 value) w/purchase.

Paradise No. ActiveX & DLL All 1D I41 0230-GL

$425.95

Paradise No. ActiveX & DLL Plus 2D I41 0211-GL

$589.95

Installation Suites

by Wise Solutions

Wise Solutions introduces a new generation of versatile installation suites for a wide range of developer needs with new features and enhancements that make installation scripting, patching, repackaging, and Web deployment a snap.

InstallMaster Paradise No. G10 0710-GL

$699.95

InstallBuilder Paradise No. G10 0610-GL

$369.95

InstallMaker Paradise No. G10 0910-GL

$179.95

TestTrack 5-User

by Seapine Software

TestTrack Simply Better Bug Tracking

TestTrack is the fastest and most complete multi-user bug tracking solution for Windows 95/NT. Tracks bug and feature requests, customers, users, test configurations, and more. Advanced features include e-mail notifications, e-mail bug import, duplicate bug handling, release note generation, and much more. Distribute TestTrack's standalone bug reporter to your customers to automate customer support. With all of its power, TestTrack remains the easiest bug tracking solution to use and maintain.

Paradise No. S96 0221-GL

$945.00

c-tree Plus®

by FairCom

DOS • WINDOWS • NT • UNIX • OS/2 • SUN • RS6000 • HP9000 • MAC • QNX • LINUX • SCO.
This well known, highly-portable data management package has become established as the tool of choice for commercial development. Offering unprecedented data control, choose from direct low level access, ISAM level, or ODBC access with the FairCom Server. Single User, Multi User, or optional Client/Server, ANSI Standard. Full Source.
No Royalties.

Paradise No. F01 0120-GL

$835.95

VTune® Performance Enhancement Environment

by Intel

The comprehensive solution for developers of high performance software for Intel® Architecture computers—from the people who design the processors.

- VTune™ Performance Analyzer
- Intel C/C++ Compiler
- Intel Performance Library Suite
- Intel Performance Training Center.

Paradise No. I23 0130-GL

$375.95

SPF/SourceEdit v.2.5

by Command Technology

SPF/SE provides ISPF style file management and editing for Win 95/98/NT

Features include: ISPF/PDF command set, source colorization; ASCII or EBCDIC character set; Variable or Fixed records; record lengths to 64K; file sizes to 100M; Hex display/modify; Mappable keys; "C" macro language; Dialog Definition Language; and much more ...

Paradise No. C29 0310-GL

$119.95

DevTrack™ 3.5

by TechExcel

DevTrack is the premier defect- and project-tracking tool that comprehensively tracks and manages all defect and development issues. DevTrack 3.5 adds exciting enterprise-level features: a definable workflow; master/detail relationships; field-level security; and the best MS Visual SourceSafe integration on the market. Intuitive and powerful, DevTrack provides an integrated client/server and intranet/Internet solution for defect management. Features universal ODBC support, extensive customization, powerful e-mail, and presentation-quality reports and graphs.

Paradise No. T34 0140-GL

$359.95

CodeWright 6.0

by Premia

New in CodeWright 6.0, the #1 selling programmer's editing system for the Windows platform:

- CodeSense—word completion and function parameter help
- CodeFolio—work faster by giving your teams a shared view of your most commonly used code
- Sync Technology support for Visual Basic 6.0 and Borland Delphi 4
- Bi-directional synchronization for Visual Studio 97 and 98.

Paradise No. P36 0120-GL

$269.95

FarPoint's Input Pro 3

by FarPoint Technologies

Easily create professional data-entry screens using Input Pro's eight custom controls. Input Pro's formatted-edit controls automatically validate date/time, numeric or text data entered by your user, database or as the application programmatically assigns or edits values. Use the boolean control to display a custom true/false/grayed value appearance. Use the memo control to display large amounts of text, even those greater than 64K.

Paradise No. F02 0610-GL

$109.95

To order—www.pparadise.com or 800-445-7899

Programmer's Paradise®

RoboHELP® Office 2000

by Blue Sky Software

The Industry Standard in Help Authoring

More software developers worldwide use best-selling RoboHELP Office to create full-featured Help and documentation systems for their applications. RoboHELP Office has won more awards than all other Help tools combined.

- Single source for Microsoft HTML Help, WinHelp, JavaHelp, WebHelp, Windows CE Help, and printed documentation
- Easy to use WYSIWYG authoring environment
- Quickly produce great online documentation
- Deliver projects on time
- Shorten the process of creating online Help
- Generate printed manuals, training guides, & more
- Y2K Compliant.

New Version

Programmer's Paradise '98 *Top Product*

Paradise No. B13 0260-GL

$889.95

True WebChart™ 6.0

by APEX Software Corporation

Easily create stunning, lightening fast charts for Active Server Page (ASP) applications with NEW True WebChart 6.0, the ultimate charting tool for browser-independent Web server applications. Serve charts and graphs as JPEG or PNG to any browser! Or serve charts as an ActiveX control to allow your end-users to change chart types and data values! True WebChart 6.0 has built-in Olectra® Chart technology to provide chart types galore; data binding to any data source without writing a single line of code; zooming, scaling, and rotation; and much more.

Paradise No. A34 0910-GL

$689.95

Visual SlickEdit

by MicroEdge, Inc.

Visual Slickedit®, the award-winning editor from MicroEdge, Inc. increases development productivity, and improves software quality. It supports most languages out of the box and is extendible to support your favorite language as well.

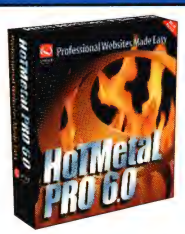
With a multi-platform presence and integration with industry leading development environments, Visual SlickEdit provides an entire organization with a standard coding environment.

Paradise No. M39 0130-GL

$285.95

HoTMetaL PRO 6.0

by SoftQuad Software, Inc.



HoTMetaL PRO 6.0, is the world's most versatile Website creation and management tool. Powerful, customizable and extensible, it gives professional developers the advanced capabilities and productivity tools needed to quickly create sophisticated sites. With unrivaled editing, site management and publishing tools, HoTMetaL PRO 6.0 is the ultimate Web development environment.

Paradise No. S32 0151-GL

$92.95

InstallShield Pro 5.5

by InstallShield Software Corp.

InstallShield® Professional is the commercial developer's choice for creating industry-standard Microsoft® Windows® application installations. Critically acclaimed and used by over ninety of the top 100 independent software vendors, InstallShield Professional offers both scripting power and timesaving visual tools in an integrated development environment. The choice for intricate application requirements is InstallShield Professional.

Paradise No. I21 0330-GL

$789.95

SourceOffSite Pro Ed. 2.0

by SourceGear

Access SourceSafe™ over the Internet

New Version

Connect to a centralized Visual SourceSafe database using any standard TCP/IP connection.

- IDE integration
- Over 10 times faster than RAS
- Efficient—uses data compression
- Secure—up to 128-bit data encryption
- Familiar—the look and feel of VSS
- Cross-Platform—98/NT/UNIX.

†† Pro edition, 1 users.

Paradise No. S0Z 0230-GL

$179.95††

LEADTOOLS Imaging

by LEAD Technologies, Inc.

LEADTOOLS is a family of comprehensive toolkits designed to help programmers integrate color, grayscale, document, medical, multimedia and vector imaging into their applications quickly and easily. Whatever your programming needs, LEAD has a toolkit specifically designed to give you the best imaging technology available with the stability and dependability you would expect from the imaging LEADer.

Paradise No. L05 0151-GL

$355.95

Microsoft Visual Studio 6.0

by Microsoft Corporation

Build scalable Windows 2000 applications today

The Microsoft® Visual Studio® 6.0 development system now provides the comprehensive resources necessary for successful Windows® 2000 development. From building e-commerce Web solutions that take advantage of the new Windows 2000 clustering server to highly scalable data-driven business applications, Visual Studio 6.0 Enterprise Edition contains everything you need in one development suite.

MS Visual Studio Pro Ver. Upgrade	
Paradise No. M47 012A-GL	REBATE! **$379.95†**
MS Visual Studio Pro Competitive Upgrade	
Paradise No. M47 012A-GL	**$479.95**
MS Visual Studio Professional	
Paradise No. M47 0129-GL	**$965.95**
MS Visual Studio Enterprise	
Paradise No. M47 022F-GL	**$1,449.00**
MS Visual Studio Enterprise Ver. Upgrade	
Paradise No. M47 022G-GL	**$965.95**

† Price after manufacturer's $100 mail-in rebate.

GUARANTEED BEST PRICES*

Should you see one of these products listed at a lower price in another ad in this magazine, CALL US! We'll match the price, and still offer our same quality service and support!

***Terms of the offer:**

- Offer good through November 30, 1999
- Applicable to pricing on current versions of software listed
- November issue prices only
- Offer does not apply towards obvious errors in competitors' ads
- Subject to same terms and conditions

Prices subject to change.
Not responsible for typographical errors.

NASDAQ: PROG

For a Free catalog—Call 800-445-7899

intel inside
pentium® III

Phone Hours: Mon-Fri 7am-9pm CT • Sat 10am-6pm CT • Sun 12pm-5pm CT
In Canada,[1] call 800-839-0148 In • Mexico,[1] call 01-800-021-4531 • GSA Contract #GS-35F-4076D

[1]Prices and specifications valid in the U.S. only and subject to change without notice. [3]On-site service may be provided pursuant to a service contract with a third-party provider and may not be available in certain areas. Technician will be dispatched if necessary following phone-based troubleshooting. [4]For hard drives, GB means 1 billion bytes; total accessible capacity varies depending on operating environment. [5]17X Min. [15]Business leasing arranged by Dell Financial Services L.P., an independent entity, to qualified customers. Above lease payments based on 36-month lease, and do not include taxes, fees, shipping charges; subject to credit approval and availability. Above lease terms subject to change without notice. 3Com is a registered trademark of 3Com Corporation. The Intel Inside logo and Pentium are registered trademarks and Pentium III Xeon is a trademark of Intel Corporation. Microsoft and Windows NT are registered trademarks of Microsoft Corporation. Other trademarks, trade names and logos are the property of their respective owners. Dell and the DELL logo are registered trademarks and Dell Precision is a trademark of Dell Computer Corporation. ©1999 Dell Computer Corporation. All rights reserved.

BE DIRECT™

DELL®

www.dell.com

The Dell Precision Workstation.
For people who believe in possibilities.

DELL PRECISION WORKSTATION 610	**starting at $2899**	**FEATURES**	**GRAPHICS CARD**	**Customize Online with E-VALUE CODE: 89298-491028**
Up to dual Pentium® III Xeon™ Processors from 500MHz–550MHz (RAID Capable)	Business Lease[15] starting at $97/Mo., 36 Months	64MB up to 2GB 100MHz ECC SDRAM 9GB[4] up to 36GB[4] (10,000 RPM) Ultra-2/LVD SCSI Hard Drives	32MB Diamond Viper V770D (Upgradeable to: Appian Jeronimo Pro, Diamond Fire GL1 or Intense3D™ Wildcat™ 4000)	
DELL PRECISION WORKSTATION 410	**starting at $2199**	**FEATURES**	**GRAPHICS CARD**	**E-VALUE CODE: 89298-491021**
Up to dual Pentium III Processors from 450MHz–600MHz (RAID Capable)	Business Lease[15] starting at $74/Mo., 36 Months	64MB up to 1GB 100MHz ECC SDRAM 9GB[4] up to 36GB[4] (10,000 RPM) Ultra-2/LVD SCSI Hard Drives	32MB Diamond Viper V770D (Upgradeable to: Appian Jeronimo Pro, Diamond Fire GL1 or Intense3D Wildcat 4000)	
NEW **DELL PRECISION WORKSTATION 220**	**starting at $1999**	**FEATURES**	**GRAPHICS CARD**	**E-VALUE CODE: 89298-491019**
Up to dual Pentium III Processors from 450BMHz–600BMHz NEW 133MHz System Bus increases data throughput/transfer rates for better system performance	Business Lease[15] starting at $67/Mo., 36 Months	128MB up to 512MB PC800 RDRAM at 400MHz 6.4GB up to 27GB EIDE or 9GB[4]–36GB[4] (10,000 RPM) Ultra-2/LVD SCSI Hard Drives	32MB 4X AGP Diamond Viper V770D (Upgradeable to: Appian Jeronimo Pro) NEW 4X AGP in combination with new chipset and RDRAM provides up to 25% improved intense graphics	

Please contact your sales representative for more information as estimated ship dates will exceed 30 days.

DELL PRECISION WORKSTATION 210	**starting at $1799**	**FEATURES**	**GRAPHICS CARD**	**E-VALUE CODE: 89298-491017**
Up to dual Pentium III Processors from 450MHz–600MHz	Business Lease[15] starting at $60/Mo., 36 Months	64MB up to 512MB 100MHz ECC SDRAM 6.4GB[4] up to 27GB[4] EIDE or 9GB[4]–18GB[4] (10,000 RPM) Ultra-2/LVD SCSI Hard Drives	32MB Diamond Viper V770D (Upgradeable to: Appian Jeronimo Pro)	

COMMON FEATURES: 17" (16.0" vis) M780 Monitor [Upgrades Available from 17" (16.0" vis) P780 to 24" (22.5" vis) Multi-Monitor Capability], Microsoft® Windows NT® 4.0, Integrated 3Com® 10/100 PCI NIC with Remote Wakeup, 40X Max[5] Variable EIDE CD-ROM Drive, Integrated Sound Blaster Pro Compatible Sound, Remote Client Manageability support via DMI 2.0, Wakeup On LAN Capable, 3-Year NBD On-site[3] Service, 7x24 Dedicated Hardware Tech Support

AWARDS: Dell Precision Workstation 610 — *PC Magazine's* Editors' Choice Award (5/99) and *Digital Video*'s Award of Excellence (3/99)

888-395-3447

TO CUSTOM BUILD YOUR OWN AND ORDER ONLINE

www.dell.com/smallbiz/dev

Dr. Dobb's News & Views

The World's Smallest Web Server

A graduate student at the University of Massachusetts, H. Shrikumar, has developed the most recent "world's smallest web server"—literally smaller than a typical fingernail. Researchers have been leapfrogging each other with ever-smaller web-server implementations, but this one will be hard to beat. The nearest competitor is a matchbox-sized 486SX implementation from Stanford University's Wearable Computing Laboratory. While the Stanford web server is basically a full-fledged single-board PC costing several hundred dollars to manufacture (see http://wearables.stanford.edu/), Shrikumar's "iPic Server" is made from parts costing a few dollars.

The key to the iPic Server is that Shrikumar managed to shrink an RFC-1122-compliant TCP/IP stack to 256 bytes. Using an 8-bit PIC microcontroller coupled with a 32-KB EEPROM, the system can fit on a circuit board about the size of a match-head. A proof-of-concept demonstration (see http://www-ccs.cs.umass.edu/~shri/iPic.html) includes several Java applets and Postscript files describing the system, all stored in 32 KB on the EEPROM. The tiny iPic Server would be ideal for controlling appliances and other simple devices via the Internet.

Shrikumar told *DDJ* that "the biggest challenge was developing the TCP/IP stack. Setting up the system took only about 15 days, but it took over four months to develop a minimal TCP/IP stack that conformed to the standard and would deliver well-behaved packets." In addition to the TCP/IP stack, Shrikumar wrote a tiny real-time operating system (essentially an RMA task-scheduler) and a simple telnet implementation for editing files on the chip.

CPU Flushed with Success

It's hard to imagine how a new microprocessor could have anything in common with public restroom toilet partitions, but Advanced Micro Devices (AMD) has managed to make the connection. AMD's new line of Athlon processors shares its name with a trademarked material called "Athlon," manufactured by Trespa North America, Ltd. According to Trespa's web site (http://www.trespa.com/), "Toilet Partitions made from Trespa Athlon® panels function well in humid areas such as around swimming pools, shower areas, or gyms." Lest there be any confusion, Athlon panels can be cleaned with soap and water, a practice not recommended for Athlon processors. The moral: Trademark searches are a good thing.

Light in the Slow Lane

The principles of quantum mechanics are leading to advances in controlling light for optical applications. The quantum optics group at Texas A&M University, led by physics professors Marlan Scully and George Welch, has come up with a way of slowing down light by a factor of 3 million using a relatively inexpensive and simple process. Heating the metal rubidium to 100 degrees Celsius, which turns it into a vapor, and then passing light through the heated gas, slows down the speed of light to about 90 meters per second. The possibility of manipulating light at such slow speeds could lead to, among other things, new methods for controlling optical signals, storing optical data, and developing optical switches.

The key to the process is that the light-absorptivity of the rubidium vapor can be controlled by applying a separate laser beam to the vapor. The laser beam changes the quantum state of the rubidium subatomic particles so that the vapor is virtually transparent, thus preserving the intensity of the light passing through the gas at the slow speed. And since the light intensity can be varied, the process can be applied to the concept of optical switching. Signal delay is another potential application for this process. The rubidium process can be achieved in a container the size of a test tube, potentially eliminating the need for extensive fiber optic cabling to produce signal delays. For more information, see http://focus.aps.org/v3/st37.html.

Windows CE Gives You Gas

Before long, pumping gas will join the ranks of just another online experience. BP Amoco, Microsoft, and gas-pump manufacturer Tokheim have teamed up to develop gas pumps running Windows CE with a web-browser terminal, from which you can check e-mail and surf the Internet while pumping gas. Whether this is a good idea remains to be seen. Still, it may be a boon for companies wanting to advertise their wares. According to *The Wall Street Journal* (August 17, 1999), the average gas pump serves about 70 people per day. Multiply that by 100,000 pumps and an ad on a pump web terminal could reach 7 million consumers.

The ACM Says No to Software Engineering Licensing

The Association for Computing Machinery (ACM) has declined to support a plan by the Texas Professional Engineers Licensing Board (http://www.main.org/peboard/) to administer a licensing exam for software engineers, as it does for traditional engineering disciplines such as mechanical and civil engineering. In a policy statement, the ACM announced that it "is opposed to the licensing of software engineers at this time because ACM believes that it is premature and would not be effective at addressing the problems of software quality and reliability." According to the policy statement, the ACM is committed to improving software quality by promoting research and development, and "identifying standards of practice." Meanwhile, the Texas licensing board has postponed the establishment of a software engineering license.

NT versus CE

Muddying the OS waters as only it can do, Microsoft has announced Windows NT Embedded 4.0—a version of NT targeting OEMs developing high-end embedded products such as integrated voice response systems, PBXs, call center systems, medical appliances, and advanced printers. The primary difference between standard NT 4.0 and NT Embedded 4.0, is that the embedded version has been "componentized," letting you pick and choose services and protocol stacks as needed.

NT Embedded can run headless, that is, without a graphical user interface, and on "spindleless storage," such as flash memory, CD-ROM, and disk-on-chip systems. It can run on 486SX or more powerful processors with a minimum of 8 MB of RAM and 12 MB of ROM. Vince Mendillo, a Microsoft product manager, told *DDJ* that NT Embedded does not compete with Windows CE, which targets lower-end embedded applications such as cell-phones, PDAs, and other devices running with as little as 40 KB of RAM and 256 KB of storage.

"I'll take Multi-Platform for $600."

The category is **SCM** *Software Configuration Management.*

t runs on over 30 platforms.

- It runs on over 30 platforms: Windows, Macintosh, every variation of UNIX you can think of, and more. And Perforce provides the same interface on all platforms.
- It routinely manages code bases of more than 100,000 files, including source, document, and Web content.
- It scales to hundreds of concurrent users.
- It works equally well over the Internet, the office LAN and the global corporate WAN.
- It controls the evolution of multiple, concurrent development and release code lines.
- It features near-zero downtime and near-zero administration.

Please phrase your answer as a question.

If you asked **"What is Perforce?"** you're on your way to **managing code development on any platforms you choose.**

Try a copy from www.perforce.com, and call us for free technical support to help with your evaluation. Don't worry about "content-free" salespeople calling you - we believe the world's fastest Software Configuration Management system speaks for itself.

For More Info
www.perforce.com

US Headquarters
Perforce Software, Inc.
2420 Santa Clara Ave.
Alameda, CA 94501 USA

+1.510.864.7400_tel
+1.510.864.5340_fax
info@perforce.com

United Kingdom
Perforce Software UK
15 Woodhurst Lane
Wokingham RG41 1JQ

+44 (0) 1189.893.200_tel
+44 (0) 1189.893.201_fax
uk@perforce.com

Don't jeopardize your software development projects. Use Perforce for SCM.

PERFORCE
THE FAST SOFTWARE CONFIGURATION MANAGEMENT SYSTEM
www.perforce.com

BM OS/390, Linux, Macintosh, Solaris, VMS, Windows 95/98, Windows NT are all trademarked.

Beware of Geeks Bearing Gifts

Plan Your e-Security Defense! Attend RSA™ Conference 2000.

Learn how to turn your business into a hacker-proof fortress at RSA Conference 2000, January 16-20, 2000 (A.D.) San Jose McEnery Convention Center, San Jose, California

To you, e-business is a major potential source of profits and perhaps the future of your enterprise. To the hackers of the world, it's a golden opportunity to pillage, plunder and destroy.

The citizens of Troy never knew what hit them. But you can prepare by attending the most important information security gathering of the year: RSA Conference 2000.

The ultimate millennium-starter for e-business warriors. The annual RSA Conference is a unique gathering of professionals in business, development, government, academia, the investment community and the media. Join us and attend sessions led by the world's foremost security developers and industry analysts, participate in tutorials and immersion training, and network with over 8,000 of your peers.

The more you know, the less you have to fear, and the farther you can take your e-business initiatives. You'll walk away understanding how information security can increase your revenues, how PKI can improve the efficiency and productivity of your organization and much more. **Remember, in the world of e-business, the best offense is a good defense.**

Each year the RSA Conference pays tribute to a culture that played a prominent role in the history of secure communications. This year's Conference salutes the ancient Greeks, who used encrypted messages to communicate on the battlefield–over two and one-half millennia ago.

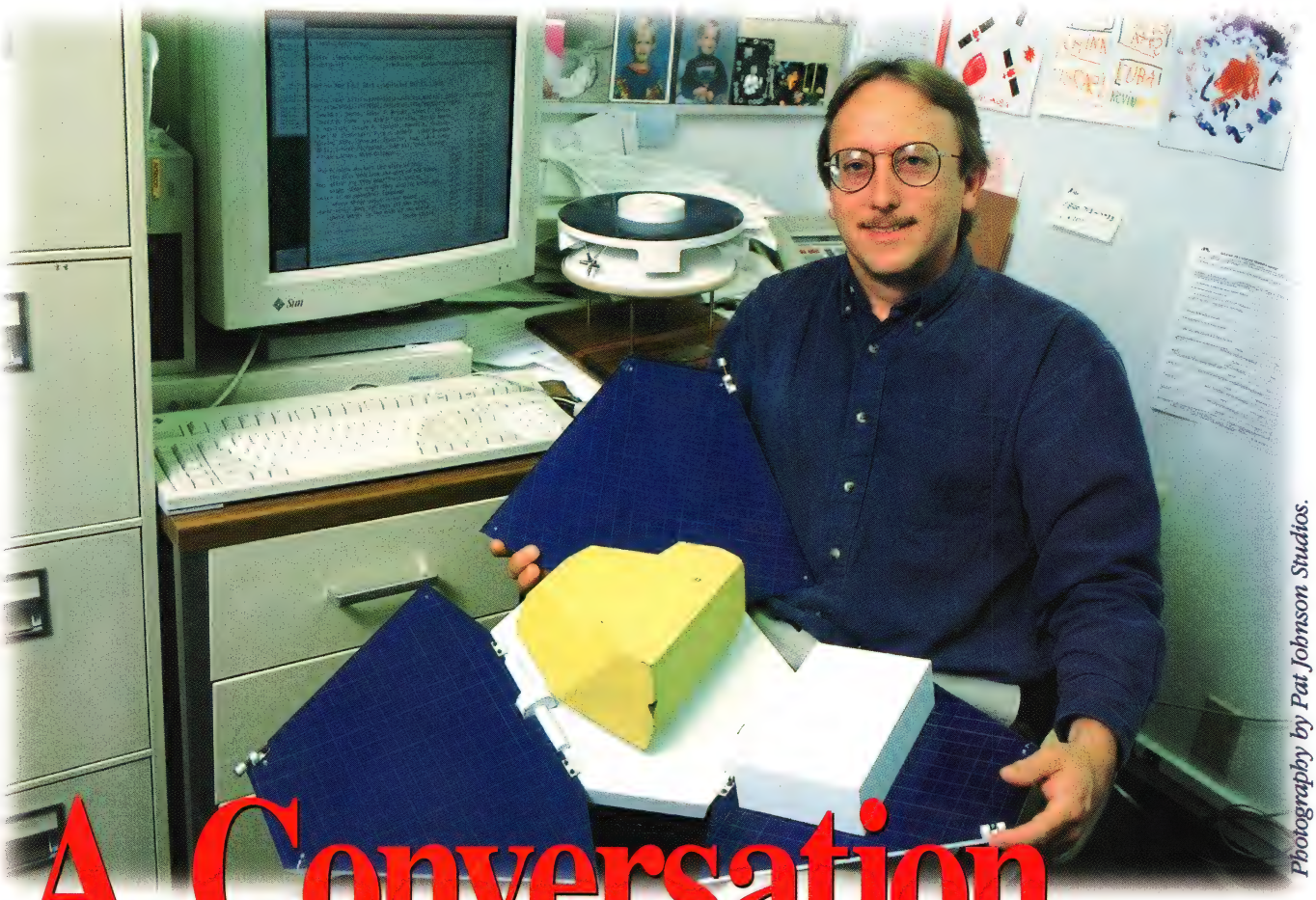
To register or obtain a free expo pass, call **1.800.340.3010 or visit www.rsasecurity.com**

RSA™ Conference 2000

Major Sponsors Include

RSA SECURITY IBM Microsoft COMPAQ intel

RSA and the RSA Security logo are trademarks of RSA Security Inc. The IBM, Compaq, Microsoft and Intel logos are registered trademarks of IBM, Compaq, Microsoft and Intel. All other trademarks are the property of their respective owners. © 1999 RSA Security Inc. All rights reserved.



Photography by Pat Johnson Studios.

A Conversation with Glenn Reeves

Really remote debugging for real-time systems

Jack J. Woehr

The Mars Pathfinder Mission was the first mission of NASA's Mars Surveyor program — a decade-long program of robotic exploration that focused on the search for evidence of past life on Mars, understanding the Martian climate and its lessons for the past and future of Earth's climate, and understanding the geology and resources that could be used to support future human missions to Mars. As such, the mission (managed by the Jet Propulsion Laboratory; http://www.jpl.nasa.gov/) was one of the most ambitious and closely watched space missions in history.

Pathfinder essentially consisted of a stationary lander (the "Lander") and surface rover (called "Sojourner"), which together had the primary objective of demonstrating the feasibility of low-cost landings on and exploration of the Martian surface. The Pathfinder system itself was built around a variety of off-the-shelf hardware and software components. At the heart of the system was a single CPU — the RS6000 — running the Wind River Systems VxWorks real-time operating system.

Pathfinder was launched on December 4, 1996, and had a seven-month cruise to Mars. It landed on Mars on July 4, 1997, where Sojourner almost immediately began conducting

Jack is a DDJ *contributing editor. He can be contacted at jax@well.com.*

experiments and collecting data; see Figure 1. In the first month of surface operations the mission returned to NASA about 1.2 gigabits of data, including 9669 Lander and 384 Rover images and about 4 million temperature, pressure, and wind measurements.

Suddenly, on September 27, 1997, communication with Sojourner was lost as the system began experiencing total resets, along with the resulting loss of data. The mainstream press promptly— and generally inaccurately— pounced on the problem, referring to "software glitches" that were due to the computer "trying to do too many things at once."

The problem, as JPL engineers like Glenn Reeves (the "Flight Software Cognizant Engineer for the Attitude and Information Management Subsystem, Mars Pathfinder Mission"), quickly discovered involved priority inversion— a phenomenon familiar to real-time operating-systems engineers for more than 20 years. Contrary to most published reports, the problem was hardly extraterrestrial: A low-priority task (such as one for meteorological data gathering) occasionally grabbed a semaphore needed by a top-priority task (a bus management task, for example), then got preempted by medium-priority tasks (for instance, a communications task). Therefore, the high-priority task was not able to complete its work by the specified time. When the blocked top-priority task didn't do its work in time, a watchdog thread reset the system. This reset reinitialized all of the hardware and software. It also terminated the execution of the current ground commanded activities. Data that had already been collected wasn't lost. However, the remainder of the activities for that day could not be accomplished until the next day.

Not only were JPL engineers aware of priority inversion from the outset, but they had carefully planned for instances of it. Consequently, they were able to quickly reproduce the problem on a duplicate system in the lab, then transfer a fix— changing the system default priority for *select()* semaphore creation— to the spacecraft about 100 million miles away. (Talk about remote debugging!) This was doable because they had originally created two writable images of the control software, and either one could be deleted and reloaded from Earth. The team chose to never delete one of the two original copies while patching from the ground, for fear of losing something clean to boot with.

In the end, all went well. The spacecraft completed its mission. Reeves, who is now the Chief Engineer, Mission Data System Project, recently took time to chat with *DDJ* contributing editor Jack Woehr about Pathfinder and other issues involved in writing software for extraterrestrial exploration.

DDJ: You're programming computers that operate on the surface of another planet. That's weird!

GR: It's weird for me, let me tell you. You're in this microcosm of people where your whole life is concentrated on making sure the next day's work of activities are going to go just right. You're walking home at 3:00AM still wired from staying up all night looking at information coming back. Your wife and the kids are asleep, there's no way you can sleep and you've go to be back at 8:00AM anyway. So you turn on CNN and there's your smiling face in the control room and you're looking at the same images you looked at 12 hours ago. It's a strange feeling.

Figure 1: *Sojourner on the surface of Mars. (Courtesy NASA/JPL/Caltech.)*

It was a small project by a small team, yet it had a phenomenal impact overwhelming to many of us. On the technical side, it seemed straightforward at the time.

DDJ: I guess when you're doing the code, it's like any other real-time project.

GR: When you have a ballistic entry, there's not much you can do about it. There's a little unknown, where you're relying on information gathered 22 years ago, but in general, we tried to simplify that job as much as we could.

We spun the spacecraft, we had inertia on our side...the control part was fairly straightforward. The hard part was thinking what things could go wrong, what do we do if something breaks...There are very few things that are redundant. How are we to go about testing it? How do we make sure that we can build an environment that matches what we want the spacecraft to fly through? Things like that turn out to be much harder than the spacecraft development itself.

DDJ: That's always a problem when one designs a large system, how one tests it in anything resembling the deployment environment.

GR: Space is pretty forgiving, few forces, everything action and reaction. Coming down through the atmosphere, that's a little different ballgame, but we can still do decent simulations. We gave up on trying to model bouncing and orientation of the vehicle and the simulation of the accelerometer values during the movements. Those were too hard to do. We tested the thing as hard as we can in a brute force sort of way.

DDJ: You dropped it from a tall building?

GR: Some drop testing. We did a lot of testing of the airbags. We proved to ourselves they could withstand the forces of the bounces, made sure they wouldn't bottom out, would retain the right pressure, that the airbag material itself wouldn't rip prematurely or catastrophically. We did that for two or three years, it was either the most expensive portion of our test program or pretty close.

The actual orientation, opening the petals and turning the vehicle over, was very much brute force. Put the thing on the ground, read the accelerometers, see that it opened the right petal in the right order.

We did testing of the airbags as they are retracted. They get pulled in by cables within the bag. We did a lot of testing with rock placement, "Given the fact that bag's going to wrap around this rock, are the motors going to stall and leave the bag extended and flapping in the breeze?"

In the end, the darn thing bounced 16 times, landed right on its base, and all the airbags retracted right in, just like it was supposed to. The perfect scenario happened in the actual mission.

DDJ: It seems almost petty to ask about the computer programming problem in comparison with the flow of the entire enterprise.

GR: In this mission, most of the navigation was done from the ground. The attitude of the spacecraft was controlled onboard.

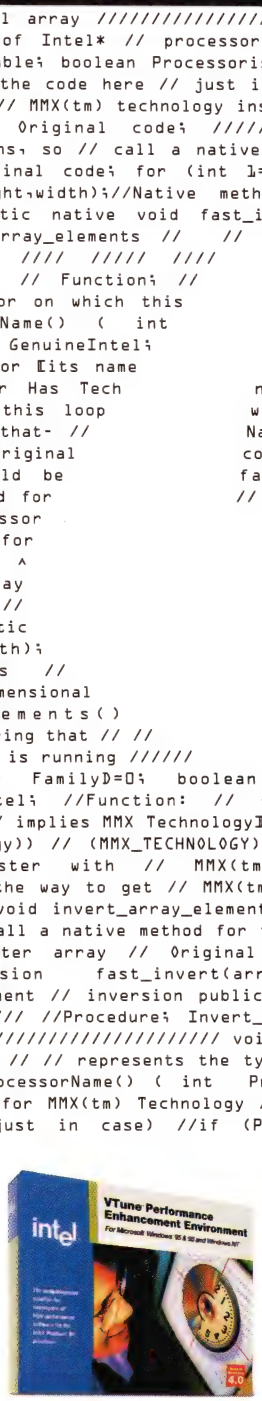
JPL used to build their own computers; in fact, at a point in time, we built our own CPUs. Not any longer, that's an art form that's better left to industry. Single-board computers are an off-the-shelf item.

So for the second time in history, JPL went out to buy a flight computer on contract. We put in a provision that specified an operating system on top of that. I wrote a lot of the flight computer specifications.

<Optimize your code>

and make it perform more powerfully // on the latest Intel® processors

[check those apps! <!!~]



//Here's the easy way to increase your code's power, especially if you're pressed for time. It's the VTune™ Performance Enhancement Environment 4.0. All the tools you need to take advantage of the power of the latest Intel processors, including the Pentium® III processor. The VTune Performance Analyzer quickly identifies performance bottlenecks and delivers priceless tuning advice. The high-performance Intel C/C ++ Compiler features support for the Pentium III processor's Streaming SIMD Extensions, and plugs into Microsoft Visual Studio.* And if that wasn't enough to whip your code into perfect shape, there's the Intel Performance Library Suite and the Intel Performance Training Center. Not surprisingly, this simple way to add extra muscle to your apps comes from the people who know the processor best. So optimize your code and make it as powerful as you imagined. < Close dialogue now >

Download a free, 14-day evaluation @ http://developer.intel.com/vtune

intel®

©1999 Intel Corporation. All rights reserved. Intel and Pentium are registered trademarks, and VTune is a trademark of Intel Corporation.
*All other brands and trade names are the property of their respective owners.

(continued from page 22)

I was very familiar with [Mentor Graphics] VRTX and [Wind River Systems] VxWorks, and have done some [Integrated Systems] pSOS work before. So I specified a fairly generic real-time operating system. What was proposed by IBM, the winner of the flight computer contract, was OSOpen, done by IBM's Raleigh, North Carolina division. It was in early beta.

We looked at it, and they had some more work to do on it. Then suddenly around 1993–1994, IBM Federal Systems got sold to Loral. What had been one company became two divisions of two large companies that were mutually antagonistic. It became obvious that we weren't going to get a real-time operating system from these guys in Raleigh within the time frame we needed it.

That's when we decided to go to Wind River and see if they would come in and port for us to this RAD6000 computer.

DDJ: The original IBM RISC mask, rad-hard?
GR: Right. In a nutshell, we sold Wind River on the neat publicity they would get porting VxWorks onto this processor. They bit, and did it fairly inexpensively and they did it quickly. We had a working version of VxWorks within four months.

Not only were they receptive to the idea, but they put people on it right away. They were very generous with time, there was no nickel-and-diming on this contract. When there were problems...I had a team of seven people doing the flight software for this (the rover was entirely separate)...the technical interchange was engineer-to-engineer, there was no contract management in the middle. It was a nice interface. They did a good job.

DDJ: And when you had to hot-patch in flight?
GR: That's standard procedure. You always build in the ability to change it.

DDJ: Just in case.
GR: Just in case, but JPL and a bunch of decent-sized companies have had the problem where you can't get all the software done in time for launch. You always make sure you build the capability to change things.

DDJ: When you send something 300,000,000 miles there's always the chance something you hadn't anticipated will happen.
GR: Well, as we said, space is forgiving, but on the surface of Mars and other places, you're right. But what we'd really like to do is build spacecraft that are much smarter about how to take care of things they don't precisely expect yet still achieve the things they are intended to do without interaction from the ground.

DDJ: Real robots! Were you an Asimov fan when you were a kid?
GR: Still am! Both a kid and an Asimov fan.

DDJ: What did your title on the Pathfinder mission—"Flight Software Cognizant Engineer"— signify?
GR: I was ultimately responsible for development and operation of flight software on the spacecraft itself. My butt on the line.

DDJ: Software project management, the final frontier! How do you guys actually get the product out?
GR: Here's my opinion on why Pathfinder worked as well as it did: We were focused. We had a specific goal, to get this

Priority Inversion: How We Found It, How We Fixed It

Glenn E. Reeves

The Mars Pathfinder spacecraft had a single RS6000-based single-board computer residing on a VME bus to control the spacecraft. The single VME chassis also contained interface cards for the radio, camera, and an interface to a 1553 bus. The 1553 bus, in turn, connected to the "cruise stage" and the "lander" part of the spacecraft. The hardware on the cruise part of the spacecraft controlled the thrusters, valves, sun sensor, and star scanner. The hardware on the Lander provided an interface to accelerometers, radar altimeter, and an instrument for meteorological science (ASI/MET). The hardware that was used to interface to the 1553 bus (at both ends) was inherited from the Cassini spacecraft.

To support the Mars Pathfinder Mission, Wind River Systems ported its standard VxWorks for the 680x0 to the RS6000. The RS6000 is the same single-chip CPU that can be found in some (now older) IBM AIX workstations. The Mars Pathfinder flight software also had several debug features that were used in the lab, but not on the actual flight spacecraft because they produced more information than we could send back to Earth. These features were not enabled, but remained in the software by design.

One of these tools was a trace/log facility that was originally developed to find a bug in an early version of the Wind River Systems VxWorks port. David Cummings, one of the JPL software engineers, built the trace/log facility. Lisa Stanley (at Wind River Systems) took this facility and instrumented the pipe services, *msgQ* services, interrupt handling, select services, and the *tExec* task. The facility initialized at startup and continued to collect data (in ring buffers) until told to stop. The facility produced a voluminous dump of information when asked.

When the "repeated reset problem" (for details, see *RISKS Digest*, Volume 19 Issue 54, January 10, 1998; http://catless.ncl.ac.uk/Risks/19.54.html) occurred on Mars, we ran the same set of spacecraft activities over and over again in the lab. Since the flight software had the trace/log facility enabled and the failing task was already coded so as to stop the trace/log collection and dump the data (even though we knew we could not get the dump in flight) for this error, we went into the lab to test whether we would have to change the software.

In less than 18 hours, we were able to repeat the problem, isolate it to an interaction of the *pipe()* and *select()* mechanisms, diagnose it as a priority inversion problem, and identify the most likely fix. In fact, the fix seemed straightforward: We had to change the creation flags for the semaphore used within the *select()* facility so as to enable priority inheritance. This change was possible because Wind River Systems supplied global variables for parameters, such as the "options" parameter for the *semMCreate* used by the select service (although this was not documented and those who do not have VxWorks

(continued on page 26)

Glenn is Chief Engineer, Mission Data System Project, at JPL. He can be contacted at Glenn.E.Reeves@jpl.nasa.gov.

com

Write programs as if they were sonnets.

Register for in-depth Java™ technology training by the true master, Sun Microsystems. We invented Java technologies, so we understand them inside and out. This means your studies will involve a more advanced approach that gets into the thick of Java technologies. Whether based in Windows or UNIX®, you will take the broadest and most rigorous courses available. Everything from the basics of Java Programming Language to the intricacies of Java Technology Architecture Planning and Design to Enterprise JavaBeans™ Programming.

And since we're continually enhancing Java technologies, you will learn the latest and greatest. In the end, you will receive the most advanced certification possible, which only Sun Microsystems can offer. For course information and enrollment, visit us at **www.sun.com/service/suned/java2**. It may be the difference between the forgettable and the truly sublime. THE NETWORK IS THE COMPUTER.™

We're the dot in .com™

Enroll at www.sun.com/service/suned/java2

©1999 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun Logo, the Java Coffee Cup Logo, Java, Enterprise JavaBeans, The Network Is The Computer and We're The Dot In .Com are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

(continued from page 24)
spacecraft launched, get it to Mars, get it through the atmosphere, deliver the rover onto the surface. We managed to focus everyone on the team on that objective.

Number two, JPL was in this Total Quality Management phase, so they talked a lot about empowerment, authority, and responsibility. Some of us, despite being more than a little skeptical about TQM itself, took that and ran with it. If we were empowered, that meant we really had the ability to make the decisions. The whole project worked that way, and the management of the project structured it. They really did trust the people working on it. You really did take responsibility for the people you put on the project.

We were the small project at JPL. Still ongoing was the back end of all the Cassini (see Figure 2) development, a 3000-person development. Pathfinder was 300 persons or fewer. So, the team was very tightly knit.

All these things contributed to the success from the management point of view. I'd love to say that JPL took that lesson and applied it to all the subsequent projects, however, I would say the exact opposite was true. A lot of people look at Pathfinder as successful, yet, "We don't want to do it that way again."

Recognize, however, that one of the reasons that's true is that, in the past, we at JPL had these huge projects and were able to build up a lot of infrastructure institution-wide within a project. But since we have a bunch of much tinier projects now, we have to build an infrastructure across the board, so there's a lot more interdependency between projects than there has been in the past.

So the Pathfinder approach of putting yourselves out, a sort of skunkworks type of environment, just doesn't work anymore at JPL.

DDJ: As space travel becomes routine, you don't need geniuses to do it.
GR: When that occurs!

DDJ: But that's your job, year after year, trying to make it routine.
GR: That's correct. And once we make it routine, we'd like to make sure that industry is doing the routine part and we go off to the stuff that's still on the frontier.

DDJ: Glenn, what are you doing now?
GR: I'm working on one of those infrastructure things. JPL is developing some avionics hardware that's radiation hardened to the point it can survive at Europa, which has a very severe radiation environment. We are pushing the technology in this arena, looking at the part level at one megarad type of problems.

This is a lot of contractual work going outside of JPL. We're trying to fly as much commercial heritage stuff as we possibly can. The flight computer we've currently selected is PowerPC 750 based at 200 MHz.

DDJ: You're putting a Macintosh into space!
GR: Isn't that convenient? The processor speed of the thing we flew on Pathfinder was about 22 MHz. We've gone up an order

(continued from page 24)
source code or have not studied the source code might be unaware of this feature). Still, the fix was not that straightforward for several reasons:

1. The code for this was in the *selectLib()* and was common for all device creations. When this global variable is changed, all of the select semaphores created after that point will be created with the new options. There was no easy way in our initialization logic to only modify the semaphore associated with the problem.
2. If we made this change and applied it on a global basis, we didn't know how it would affect the behavior of the rest of the system.
3. Because Wind River Systems deliberately left the priority inversion option out of the default *selectLib()* service for optimum performance, we didn't know if performance would degrade if we turned the priority inversion on.
4. Finally, we didn't know if there was some intrinsic behavior of the select mechanism itself that would change if the priority inversion was enabled.

In the end, we modified the global variable to include the priority inversion. This corrected the problem. We asked the Wind River Systems engineers to analyze the potential impacts for (3) and (4) above. They concluded that the performance impact would be minimal and that the behavior of *select()* would not change so long as there was always only one task waiting for any particular file descriptor (this was true in our system). I believe that the debate at Wind River Systems still continues over whether the priority inversion option should be on as the default. As for the aforementioned (1) and (2), the change did alter the characteristics of all of the select semaphores. We concluded, both by analysis and test, that there was no adverse behavior. We tested the system extensively before we changed the software on the spacecraft.

We weren't able to catch the problem before launch because the problem would only manifest itself when ASI/MET data was being collected and intermediate tasks were heavily loaded. Our before-launch testing was limited to the "best case" high data rates and science activities. The fact that data rates from the surface were higher than anticipated and the amount of science activities proportionally greater served to aggravate the problem. We did not expect nor test the "better than we could have ever imagined" case.

We did see the problem before landing, but could not get it to repeat when we tried to track it down. It was not forgotten nor was it deemed unimportant. Yes, we were concentrating heavily on the entry and landing software. Yes, we considered this problem lower priority. Yes, we would have liked to have everything perfect before landing. However, I didn't see any problem, other than that we ran out of time to get the lower priority issues resolved.

We did have one other thing on our side—we knew how robust our system was because that is the way we designed it. We knew that if this problem occurred, we would reset. We built in mechanisms to recover the current activity so that there would be no interruptions in the science data (although this wasn't used until later in the landed mission). We built in the ability (and tested it) to go through multiple resets while we were going through the Martian atmosphere. We designed the software to recover from radiation induced errors in the memory or the processor. The spacecraft would have even done a 60-day mission on its own, including deploying the rover, if the radio receiver had broken when we landed. There were a large number of safeguards in the system to ensure robust, continued operation in the event of a failure of this type. These safeguards allowed us to designate problems of this nature as lower priority. We had our priorities right.

Did we (the JPL team) make an error in assuming how the select/pipe mechanism would work? Probably. But there was no conscious decision to not have the priority inversion enabled. We just missed it. There were several other places in the flight software where similar protection was required for critical data structures and the semaphores did have priority inversion protection. A good lesson when you fly commercial off-the-shelf stuff—make sure you know how it works.

DDJ

of magnitude. That's a phenomenal increase. JPL has, historically, flown processor technologies that are 10 to 12 years behind...

DDJ:...As you wait for the rad-hard version of the part.
GR: Yup. What I'm specifically working on is a project that is sort of mislabelled as "Mission Data System." It's really an institution-wide effort to do a couple things. JPL has had several autonomy-oriented things occurring. We're trying to come up with an architecture that builds a foundation where we can have much more autonomous spacecraft than we've had in the past.

There are a lot of things involved with that, not just software, but also process, normal, up-to-date software engineering practices. I think JPL is moving from a hardware-centric organization to a more software-centric organization. The computer systems that we're building now look a lot like little local area networks. Computer systems both fast and less fast connected by things like FireWire and I2C.

DDJ: I2C is still going strong!
GR: It's not fast, but it's low power. That's often a real determinant. For high volume, high speed, we have 1394 FireWire—100 MBit/sec.

DDJ: I guess the NASA environment is pretty "toy-rich" for the programmer?
GR: We're trying to make it toy-rich in the sense that we're pushing hard for the commercial, off-the-shelf standardized things.

On Pathfinder we said, "We're going to fly the VME bus, because look at all the cards we can buy instead of doing in-house design and building every piece of test equipment."

We're still following that path.

DDJ: We started discussing the challenge of your current assignment in the context of trying to preserve the best of the Pathfinder skunkworks methodology in the current horizontal arrangement of smaller teams and standardized parts. How are you addressing that challenge?
GR: JPL is in the throes of trying to address team organization and interdependency between teams effectively. It's a big problem for an organization to go from a very command and control-oriented to a very interdependency-oriented structure.

DDJ: Is it left to JPL internally, or are you one eddy in the NASA river?
GR: At this point it's pretty much JPL internal. Our interdependency issues are certainly known to NASA.

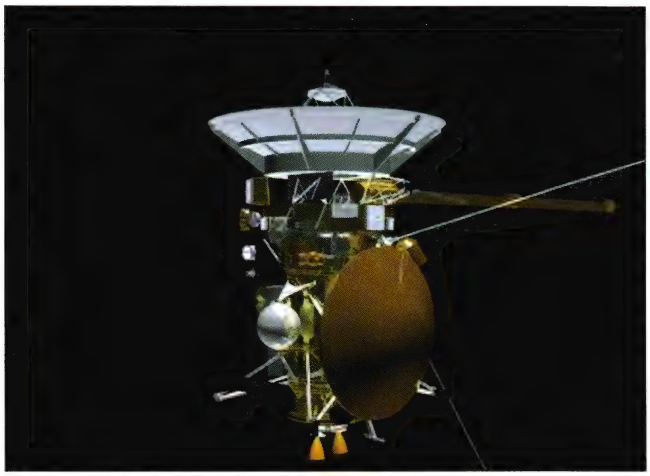


Figure 2: *Computer generated image of the Cassini spacecraft. (Courtesy NASA/JPL/Caltech.)*

FOR SOME,

ENABLING TEAM & PARALLEL DEVELOPMENT

TAKES REWRITING CODE,
FIXING THE SAME BUG TWICE,
SENDING THOUSANDS OF E-MAILS
AND RE-BUILDING
PROJECTS.

FOR OTHERS, THERE'S

StarTeam™

ONLY STARTEAM BRINGS PREDICTABILITY AND MANAGEMENT TO YOUR DEVELOPMENT PROCESS!

Getting all your technical resources **in balance** should not make your application development process more complicated and unpredictable. Fortunately, there's StarTeam, an integrated suite of products designed to increase your team's productivity, collaboration and project control. StarTeam delivers the easiest to use version control system together with an integrated project-based incident management system. StarTeam's multi vendor IDE integrations allow your developers to focus on what they do best, **coding.**

StarBase®

IT PAYS TO PREDICT THE OUTCOME™

To team enable your development projects download your free evaluation at **www.starbase.com** or call **1.888.STAR700.**

©1999 StarBase Corporation. StarBase and StarTeam are registered trademarks of StarBase Corporation. All other brands and products are trademarks or registered trademarks of their respective holders.
4 Hutton Centre Drive, Suite 800 • Santa Ana, California 92707, USA • Phone: 714.445.4400, Fax: 714.445.4404.

DDJ: You personally are being slurped into management. Do you get to write code anymore these days?
GR: Not these days...I have a position called Mission Data System Chief Engineer. It sounds like a technical decision making position, but it's primarily a programmatic activity. I'm resisting as best I can!

DDJ: All the techies reading *DDJ* are cheering, "Don't give in to the Dark Side!"
GR: Someone has to put the plan together. That deficiency has dragged me to the dark side.

DDJ: Yes, really, the management is the final frontier. Development environments are shrink-wrap now. Figuring out how to divide work suitably between people, how to interleave so the product arrives on time...
GR: Those are second-order effects. The first-order effects are methodology. UML or not UML? Languages, classic things, the ability to do things on the type of computers we're now buying...Some of the things we had to worry about in the past we don't have to worry about any more. A 200-MHz PowerPC is about 200 times faster than the processors that run in Cassini. A lot of efficiency issues are starting to disappear.

DDJ: So you just write "good old code" like anybody writes?
GR: Well, wouldn't that be nice? Wouldn't it be nice if we could get close to the maturing edge of some software engineering? JPL, because we've been relegated to targets about 10 to 15 years back, we ended up having to use whatever's available, but we're no longer in that ballgame.

DDJ: What about Linux?
GR: We haven't evaluated it from a flight perspective. It has a fairly decent presence here on desktop computers. Every engineer has his opinion on the One True Way. Languages turn out to be a religious issue, too.

DDJ: What religions are sweeping JPL?
GR: Most of the work JPL has done has been primarily in procedural languages. We did a fair amount of Ada on Cassini, but from a true object-oriented perspective, we haven't been there yet, not on the flight side. Pathfinder and others are all C-based.

But moving towards an object-oriented direction takes you to an object-oriented language, C++ or Java...We have a whole bunch of artificial intelligence folks for whom LISP is a language of choice. Factor in a multitasking, preemptive real-time environment, even a multilanguage environment, and you have a whole bunch of religious conflicts.

DDJ: You're going to need memory to run that stuff.
GR: Look at the memory densities we're starting to see. On Pathfinder we had 128 megabytes of RAM. We moved into the realm where my desktop machine has more than the spacecraft. We're continuing in that direction. Let's put it this way, the software guys will be able to use all of that space. "If it's there, we can fill it up."

DDJ: When are we going to Europa?
GR: The hardware and the software come together in the two-year time frame for the two missions that will be supported. The first one is a mission called "Space Technology 4," or ST4. It's going off to a comet [to] take some samples [and] possibly eventually return. I'm not sure about the return part now, they've done some descoping lately.

The second mission is to Europa. They're launching possibly in 2003. There are other ongoing missions that will also probably use this technology, but those are the only two signed up now.

Glenn Reeves

DDJ: What keeps people going to Congress for funding to shoot tin cans at Europa? Why does this happen?
GR: My personal opinion is that you have to recognize this is a very small planet. At some point in the future, man is going to move off into space, there's no doubt about it. People recognize there has to be some level of exploration that's ongoing to support that goal.

Perhaps the question is, "How can we spend billions in space while there are starving children in the world?" I think it's a balance, where you're balancing the short-term goals with the long-term goals.

I can't speak for the manned exploration side of NASA, but we on the robotic exploration side have witnessed great changes, going from billion-dollar spacecraft to $200 million spacecraft. NASA applies fairly small resources to greater and greater discovery and exploration missions.

DDJ: I grew up with Robert Heinlein's books about mining the asteroids. Is that in the foreseeable future?
GR: That's interesting, because this week or next week there's a guy coming from a company called SpaceDev Inc., to talk to us about commercial deep-space exploration. How there will be a shift from government-sponsored space exploration, whether Russian, Japanese, German, French, or American. I'm not sure he's going to address mining, but he's going to address that there are institutions both research and commercial that are willing to pay for images and spectroscopy in the hopes of discovering minerals, resources...

DDJ: The classic SciFi goals of space exploration.
GR: I can tell you that it's coming. Any time soon? Maybe in our lifetime, late...My kids will see it.

DDJ: Any advice for software engineers interested in space?
GR: My brother-in-law says, "Ya write the software but ya never actually know about the thing it's doing." I think that's wrong. I think you can still be a computer scientist and do phenomenally interesting applications of that art. Pathfinder is an example of how exciting that can really get.

DDJ

The name is new. The reputations are legendary.

SECURITY

For almost two decades, businesses have trusted ***Security Dynamics Technologies, Inc.*** and ***RSA Data Security, Inc.*** to secure e-business, even before it was called e-business. Today, the companies have unified under one name, RSA Security Inc. Our new name and look reflects our singular, continued commitment to innovative products that help customers meet the security needs of e-business, and the support and services that tie it all together. You may not have known us by name, but you should. Chances are you've already relied on one or more of our products to purchase something over the Internet, securely send email, safely connect to your network, or do your banking online. As a combined company, we are the most trusted name in e-security — bar none.

More than 450 million copies of our RSA BSAFE® encryption technology are installed in today's most successful applications and devices. More than half of the Fortune 100 already use our award-winning RSA SecurID® user authentication systems to protect their networks and vital information. And our new RSA Keon™ product line provides companies with a complete digital certificate system, known as "PKI," to enable and manage security for e-commerce applications. Thousands of customers trust RSA Security, including Cisco, Compaq, Ericsson, Fidelity, IBM and Lucent. Shouldn't you? To learn how we might serve your e-security needs, please visit us at **www.rsasecurity.com,** or contact us at **info@rsasecurity.com** or 1-877-RSA-4900.

The Most Trusted Name in e-Security

BSAFE, SecurID and Security Dynamics are registered trademarks, and Keon, RSA and The Most Trusted Name in e-Security are trademarks of RSA Security Inc. ©1999 RSA Security Inc. All rights reserved. All other products or companies are marks of their respective companies.

FRENCH: The language of love.

VISUAL BASIC: The language of just about everything else.

While it may not be suitable for wooing a potential love interest, the Visual Basic® language is capable of more than you realize. For instance, Microsoft® Visual Basic for Applications is embedded in more than 100 popular apps and can be used to customize and integrate them to create new solutions. Visual Basic Scripting Edition is the optimal choice for client- and server-side automation. And, Visual Basic 6.0 development system just happens to be the most productive tool for building applications for Windows® and the Web.

A romance language it is not. But in today's world, it's much more practical. To see how the Visual Basic language, in all its forms, can do everything you need to do, visit

msdn.microsoft.com/vbasic/vblang

Where do you want to go today?® **Microsoft**

© 1999 Microsoft Corporation. All rights reserved. Microsoft, Visual Basic, *Where do you want to go today?*, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Linux, Real-Time Linux, & IPC

FIFO versus shared memory

Frederick M. Proctor

When dealing with real-time systems, the overhead of interprocess communications (IPC) becomes important. The formalized structures that are used in Linux for IPC can carry with them a significant amount of overhead. This can create timing problems for your applications. In this article, I'll examine two of the best IPC mechanisms available under Linux—FIFOs and shared memory.

Communication between Linux and Real-Time Linux (RT-Linux) processes is usually accomplished via first-in-first-out (FIFO) connections—point-to-point queues of serial data analogous to UNIX character devices. FIFOs have the following characteristics:

- FIFOs queue data, so no protocol is required to prevent data overwrites.
- Boundaries between successive data writes are not maintained.
- Applications must detect boundaries between data, particularly if data is of varying size.
- FIFOs support blocking for synchronization. Processes need not poll for the arrival of data.
- FIFOs are a point-to-point communication channel. FIFOs do not support the one writer/several readers model.

Fred is a member of the Intelligent Systems Division at NIST. He can be contacted at frederick.proctor@nist.gov.

You declare the maximum number of FIFOs in rt_fifo_new.c as *#define RTF_NO 64*. These appear as devices dev/rtf0..63 in the filesystem. This limit can be changed and the rt_fifo_new.o module recompiled. There is no limit to the number of FIFOs an application can use, or to the size of data that can be written to a FIFO (within practical memory limits).

An alternative to FIFOs is shared memory, in which a portion of physical memory is set aside for sharing between Linux and real-time processes. In a nutshell, shared

memory is a pool of memory segments reserved at boot time. These segments may be mapped into the address space of more than one application, allowing for fast data sharing, data updates, and handshaking. This allows for low-latency parallel updates.

Shared memory has the following characteristics:

- It does not queue data written to it. Applications requiring handshaking must define a protocol to assure data is not overwritten.
- Because data is not queued, individual items in data structures (megabytes in size) can be quickly updated.
- It has no point-to-point restriction. Shared memory can be written or read by any number of Linux or real-time processes.
- The number of independent shared memory channels is limited only by the size of physical memory.
- Blocking for synchronization is not directly supported. To determine if data is new, the data must contain a count that can be compared against previous reads.
- Mutual exclusion of Linux and RT-Linux processes is not guaranteed. Interrupted reads and writes can be detected, however.

Whether you use FIFOs or shared memory depends on the application's natural communication model. For control applications involving processes that execute cyclically based on the expiration of an interval timer (where data queuing is the exception rather than the rule), shared memory is a good choice for communication.

Setting up the Shared-Memory Pool

The shared-memory pool is a block of physical memory set aside at boot time so that Linux does not use it for processes. To set up the pool, you first determine how much physical memory the system has and how much is to be used for shared memory. Subtracting the size of the shared memory desired from the size of physical memory gives the absolute base address of the pool. The result is passed to the Linux boot loader (LILO) at boot time. This is accomplished by editing /etc/lilo.conf and inserting a line with the *append* keyword.

Tampa, Fl	Oct. 13,14
San Jose, CA	Oct. 15
Irvine, CA (Orange Co.)	Oct. 18,19
New York, NY	Oct. 19
San Mateo, CA	Oct. 21
Columbus, OH	Oct. 26
Laguna Hills, CA	Oct. 26
Dallas, TX	Oct.*25,26
Philadelphia, PA	Oct. 27
Houston, TX	Oct.*29
New York, NY	Nov.*29,30

Pervasive Software Developer Solutions Seminar

Learn how to save hours in Web development time—FREE

Join us for a FREE one-day forum focusing on Pervasive's revolutionary new Web development tool, *Tango 2000*, and our zero-maintenance database solution, *Pervasive.SQL 2000*.

You'll learn to deliver products to the web faster, discover the latest product offerings, and even create your own e-commerce solution.

Tango 2000 is the premier open-architecture Web development tool, that delivers full front-to-back presentation, application and back-end support.

Pervasive.SQL 2000 is a complete and maintenance-free application development solution for any project or business need.

This forum is your exclusive opportunity to get hands-on knowledge of these powerful, time-saving tools—and we're bringing it to a location near you.

*Half day seminars

Hurry! Seating is limited!
For more details and registration information, go to **www.pervasive.com/devforum**

PERVASIVE® SOFTWARE

The Freedom to Create Applications for Everyone, Everywhere

(continued from page 32)

Suppose, for example, the system has 32 MB of memory and 1 MB is to be used for the shared-memory pool. The base address for shared memory is 32 MB–1 MB=31 MB. Assuming the original /etc/lilo.conf file contained Example 1(a), the file should be modified like Example 1(b).

Similarly, suppose the system has 16 MB of memory and 512 KB are to be used for the shared-memory pool. The base address for shared memory is 16384 KB–512 KB=15872 KB. The /etc/lilo.conf file should be modified like Example 1(c).

The size of the shared-memory pool must be less than the page size declared in /usr/include/asm/param.h. In Intel Pentium-class machines and above, the page size is 4 MB; on earlier machines, the page size is 1 MB.

Addressing the Shared-Memory Pool in C Programs

The base address of the shared-memory pool needs to be declared in C so that both Linux and RT-Linux code can reference it. For example, you can access shared memory based at 31 MB using the C statement *#define BASE_ADDRESS (31 * 0x100000)*. Similarly, shared memory based at 15872 KB may be accessed using the C statement *#define BASE_ADDRESS (15872 * 0x400)*. This address is used differently in Linux than in RT-Linux. Linux processes need to map this physical address into their virtual address space. RT-Linux processes can reference data located at this address as a pointer directly.

In addition to this declaration, the Linux and RT-Linux C code must agree on the data structures written into shared memory. In Listing One, for example, a Linux process sends a command to an RT-Linux process by filling in the MY_COMMAND structure and writing it into shared memory. The RT-Linux process reads this structure from shared memory to get the command. An RT-Linux process sends status to a Linux process by filling in the MY_STATUS structure and writing it into shared memory. The Linux process reads this structure from shared memory to get the status.

The MY_STRUCT structure is a combination of both the command and status structure, and can be used to ensure that the two structures do not overlap and that their fields are aligned on the proper boundaries. It is also possible to define two base addresses — one for each structure — making sure the start of one structure is after the end of the previous one and that all fields are properly aligned. By combining them into a single aggregate structure and letting the compiler allocate storage, the structure will have a valid byte alignment automatically.

In a typical application, the BASE_ADDRESS declaration and shared-structure declarations would be put in a header file shared by both Linux and RT-Linux code.

```
(a)
image=/boot/zImage
label=rtlinux-0.6
root=/dev/hda2
read-only

(b)
image=/boot/zImage
label=rtlinux-0.6
root=/dev/hda2
read-only
append="mem=31m"

(c)
image=/boot/zImage
label=rtlinux-0.6
root=/dev/hda2
read-only
append="mem=15872k"
```

Example 1: *(a) Original /etc/lilo.conf file; (b) modified file for 31 MB; (c) modified file for 15872 KB.*

Accessing the Shared-Memory Pool from Processes other than RT-Linux

Normal Linux processes are required to map physical memory into their private address space to access it. To do this, the Linux processes calls *open()* on the memory device /dev/mem; see Example 2.

Due to security concerns, the default permissions on /dev/mem allow only root processes to read or write /dev/mem. To access physical memory, the program must be run as root, its permissions must be changed to *setuid root*, or the permissions on /dev/mem must be changed to allow access to users other than root.

After the file descriptor is opened, the Linux process maps the shared memory into its address space using *mmap()*; see Listing Two. BASE_ADDRESS is passed to *mmap()*, which returns a pointer to the shared memory as mapped into the Linux process's address space. Once the shared memory is mapped, it may be accessed by dereferencing the pointer, as in Example 3(a). When the process terminates, you use *munmap()* to unmap the shared memory by passing the pointer and the size of its object; see Example 3(b).

```
#include <unistd.h>     /* open() */
#include <fcntl.h>      /* O_RDWR */
int fd;
if ((fd = open("/dev/mem", O_RDWR)) < 0)
{
  /* handle error here */
}
```

Example 2: *Calling* open() *on the memory device /dev/mem.*

Accessing the Shared-Memory Pool from RT-Linux

Shared-memory access is much easier in RT-Linux since the real-time code executes in kernel space and thus is not required to map physical addresses to virtual addresses. In Linux kernels 2.0.XX, the pointer can be set directly; see Example 4(a). In Linux kernels 2.1.XX, the pointer needs to be mapped via a call to the *__va()* macro defined in /usr/include/asm/page.h, as in Example 4(b).

Detecting New Writes

FIFOs have an advantage over shared memory in that reads and writes follow standard UNIX conventions. Processes other than RT-Linux processes can use *write()* to queue data onto a FIFO, and *read()* returns the number of characters read. Zero characters read from a FIFO means no new data was written since the last read. On the RT-Linux side, a handler is associated with a FIFO that is invoked after a nonreal-time Linux process writes to the FIFO. Normally the handler calls *rtf_get()* to dequeue the data from the FIFO.

These functions are not necessary with shared memory. Reads and writes are accomplished by reading and writing directly to pointers. Consequently, the operating system provides no way to detect if the contents of shared memory have been updated. You need to set up this handshaking explicitly.

One way to do this is to use message identifiers that are incremented for each new message. The receiver then polls the shared-memory buffer and compares the current identifier with the previous one. If they are different, a new message has been written. Handshaking to prevent message overrun is implemented by echoing message identifiers in the status structure once they have been received. New messages are not sent until the status echoes the message identifier.

This presumes a time-cyclic polling model on the part of both the Linux and RT-Linux processes. If this is not the natural model for the application, then shared memory may not be a good choice for communication. If shared memory is required for other reasons, and polling is not desirable, other synchronization between Linux and RT-Linux processes can be used. For example, RT-Linux FIFOs can be used only for their synchronization properties. A byte written to a FIFO can be used to wake up a Linux process blocked on a read, or to call the handler in an RT process.

Realizing Mutual Exclusion

It is possible (and therefore a certainty) for a Linux process to be interrupted by a real-time process while in the middle of

Have you spent the last six months mastering code with a 12-month life expectancy? The only way to know for sure is by logging on to ComputerJobs.com. We're the only site created by and for IT professionals that gives you the latest industry news, career information and job opportunities. We're the only one that's tailored to the needs of specific skillsets, like Windows developers. In fact, you can personalize your very own version of our site, which we update every hour, so you can quickly and accurately outline a plan for your future. And avoid becoming a statistic.

computerjobs.com™

Don't just upgrade your job, upgrade your future.

© 1999 ComputerJobs.com, Inc.

(continued from page 34)
a read/write to memory they are sharing. If the Linux process is interrupted during a read, the Linux process sees stale data at the beginning and fresh data at the end. If the Linux process is interrupted during a write, the real-time process sees fresh data at the beginning but stale data at the end. Both problems are fatal in general.

The problem of ensuring consistency of data shared between two processes is the subject of operating-systems research, and general solutions exist (see, for instance, *An Introduction to Operating Systems*, Second Edition, by Harvey M. Dietel, Addison-Wesley, 1990). Our problem is simpler because only the Linux process can be interrupted. In no case can a Linux process interrupt a real-time process during the execution of its task code.

This simplification means that an *inuse* flag can be used by the Linux process to signal that it is accessing the shared memory. You declare the *inuse* flag at the beginning of the shared-memory structure, and it is set by the Linux process when it wants to read or write and cleared when the Linux process is finished. The real-time process checks the *inuse* flag before it accesses shared memory; if set, the process defers the read or write action until it detects that the flag is cleared.

This may lead to the indefinite postponement of the real-time process, if the following conditions are true:

- The real-time process runs at the period of the Linux process, or at multiples of this period.
- The accesses are synchronized so that the real-time process always interrupts the Linux process during the critical section when it has set the *inuse* flag.

The first condition implies that the real-time code is running as slow or slower than the Linux code, and the second implies that the Linux code is running as deterministically as the real-time code. Neither is typically true, and both are rarely true at the same time. If these conditions are true for a system, successive deferrals can be detected by the real-time code and can trigger actions to keep the system under control.

Listing Three illustrates the application of the *inuse* flag for commands written by the Linux process to the real-time process, and status written by the real-time process and read by the Linux process. Assume that *command_ptr* has been set up in a Linux process to point to the shared-memory area for commands to the real-time process. To write to shared memory, you must compose the command, set the *inuse* flag in shared memory directly, write the command, and reset the *inuse* flag; see Listing Four.

Assuming that the real-time code has set *command_ptr* to point to the shared memory for commands, it would read commands as in Listing Five. To read status information, the Linux process sets the *inuse* flag before copying out the data. Assuming that *status_ptr* has been set up in a Linux process to point to the shared-memory area for status from the real-time process, this would look like Listing Six.

When writing status, the real-time process checks for the *inuse* flag and defers a status write if it is set. Assuming that the real-time code has set *status_ptr* to point to the shared memory for status, this would look like Listing Seven.

Queuing Data in Shared Memory Using Ring Buffers

While shared memory is most naturally suited for communications in which data overwrites the previous contents, queuing can be set up using ring buffers. Ring buffers queue 0 or more instances of a data structure, up to a predetermined maximum.

To illustrate the use of ring buffers, consider a system that queues error messages. Errors are declared as strings of a fixed maximum length, and there is a fixed maximum number of errors that can be queued. This is implemented as a two-dimensional array, as in Listing Eight.

Supplementing the actual list of errors are indices to the start and end of the queue, which wrap around from the end of the shared-memory area to the beginning (hence the name "ring buffer"), and a count of the errors queued. As previously described, an *inuse* flag is also declared to signal that a Linux process is accessing the ring buffer to prevent data inconsistencies in the event a real-time process interrupts Linux process access. The full shared-memory structure declaration is then Listing Nine.

Both Linux and RT-Linux use the same access functions. However, Linux processes need to set the *inuse* flag before getting an error off the ring, and real-time processes need to check the *inuse* flag and defer access until the flag is zero. Assuming that *errlog* is a pointer to the shared-memory area for both Linux and real-time processes, the access functions look like Listing Ten.

For Linux, getting an error off the ring buffer is accomplished by Listing Eleven. For an RT process, writing an error to the ring looks like Listing Twelve.

Sample Code

I've included a sample application that illustrates Linux to real-time commands, real time to Linux status, and real time to Linux error logging.

The application consists of two parts. The first is a real-time process that runs cyclically, reads a command buffer, continually updates a status buffer, and logs some diagnostic messages to the queued error buffer. The second part is a command-line Linux program that handles a few keyboard commands for sending commands and printing the real-time process status and error log.

This application is available electronically; see "Resource Center," page 5. The application is provided as a ZIP file and a gzipped tar file (shmex.tgz). Unpack and compile with:

```
tar xzvf shmex.tgz
make
```

You have to set up Linux to boot with shared memory set aside, as detailed previously.

Acknowledgments

Thanks to Rich Bowser, bowser@luz.cs.nmt.edu, for information on the 4-MB memory chunking and the problems with 486 systems; Albert D. Cahalan, acahalan@cs.uml.edu, for help with memory page and *mmap()* information; and Daniele Lugli, danlugli@tin.it, for review and comments.

DDJ

(Listings begin on page 38.)

(a)

```
ptr->command.arg1 = 1;
ptr->command.arg2 = 2;
```

(b)

```
#include <sys/mman.h>
#include "myheader.h"
munmap(ptr, sizeof(MY_STRUCT));
```

Example 3: *(a) Dereferencing the pointer; (b) passing the pointer and the size of its object.*

(a)

```
#include "myheader.h"
MY_STRUCT *ptr;
ptr = (MY_STRUCT *) BASE_ADDRESS;
ptr->command.arg1 = 1;
ptr->command.arg2 = 2;
```

(b)

```
ptr = (MY_STRUCT *) __va(BASE_ADDRESS)
```

Example 4: *(a) In Linux kernels 2.0.XX, the pointer can be set directly; (b) in Linux kernels 2.1.XX, the pointer needs to be mapped via a call to the* __va() *macro.*

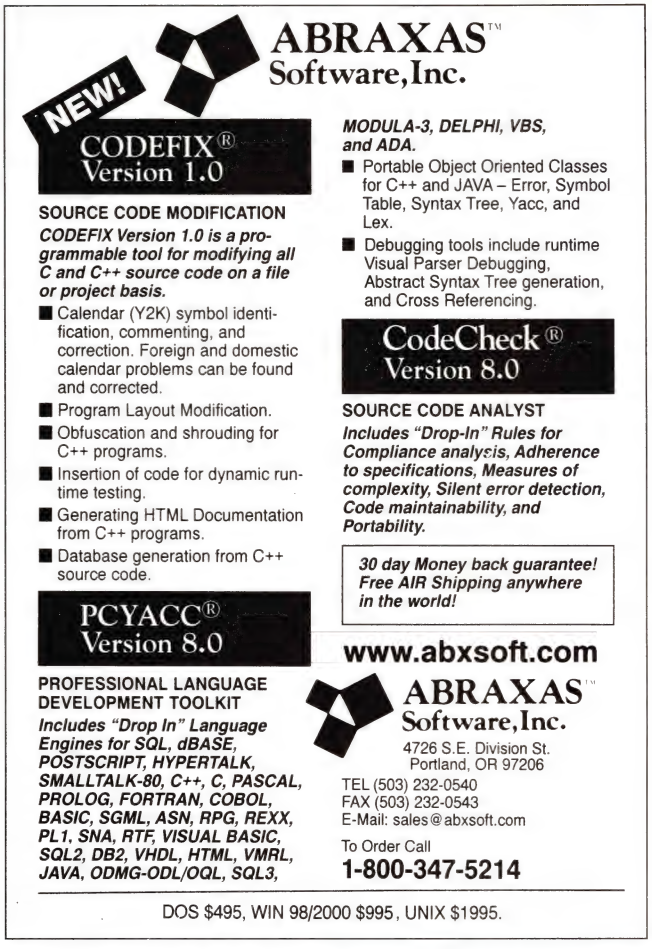
"Feeble humans!" Simon snickered. "Little do they realize I've found a way to report to my canine allies without leaving the comfort of this VB environment!"

With Seagate Crystal Reports™, you can create reports without leaving Microsoft® Visual Basic®. In fact, when used with VB 5.0 or later, the Report Designer Component gives you control to view and modify presentation-quality reports using familiar VB code. Distribution is simplified with a small component count and free run time, and you can re-use your existing Crystal Reports files. To see the latest generation of Crystal Reports, poke your head into our web site.

Access. Analyze. Report. Share.™ Seagate Crystal Reports www.seagatesoftware.com/vb

Seagate Software

©1999 Seagate Software, Inc. All rights reserved. Seagate, Seagate Software, Seagate Crystal Reports, and the Seagate logo are trademarks or registered trademarks of Seagate Software, Inc. and/or Seagate Technology, Inc. and protected by mean Rottweilers. Microsoft and Visual Basic are registered trademarks of Microsoft Corporation in the U.S. and/or other countries.



Listing One

```
typedef struct
{
 unsigned char inuse;  /* more on this later */
 int command;
 int command_number;
 int arg1;
 int arg2;
} MY_COMMAND;
typedef struct
{
 unsigned char inuse;  /* more on this later */
 int command_echo;
 int command_number_echo;
 int stat1;
 int stat2;
} MY_STATUS;
typedef struct
{
 MY_COMMAND command;
 MY_STATUS status;
} MY_STRUCT;
```

Listing Two

```
#include <stdlib.h>                /* sizeof() */
#include <sys/mman.h>              /* mmap(), PROT_READ, MAP_FILE */
#define MAP_FAILED ((void *) -1)   /* omitted from Linux mman.h */
#include "myheader.h"
MY_STRUCT *ptr;
ptr = (MY_STRUCT *) mmap(0, sizeof(MY_STRUCT),
 PROT_READ | PROT_WRITE, MAP_FILE | MAP_SHARED, fd, BASE_ADDRESS);
if (MAP_FAILED == ptr)
{
  /* handle error here */
}
close(fd);                         /* fd no longer needed */
```

Listing Three

```
typedef struct
{
  unsigned char inuse;
  int command;
  int command_number;
  int arg1;
  int arg2;
} MY_COMMAND;
typedef struct
{
  unsigned char inuse;
  int command_echo;
  int command_number_echo;
  int stat1;
  int stat2;
} MY_STATUS;
```

Listing Four

```
MY_COMMAND my_command;
/* compose command in local structure */
my_command.inuse = 1;  /* will overwrite during copy, so set here too */
my_command.command = 123;
my_command.command_number++;
my_command.arg1 = 2;
my_command.arg2 = 3;
/* set inuse flag */
command_ptr->inuse = 1;
/* copy local structure to shared memory */
memcpy(command_ptr, &my_command, sizeof(MY_COMMAND));
/* clear inuse flag */
command_ptr->inuse = 0;
```

Listing Five

```
if (0 != command_ptr->inuse)
{
 /* ignore it, perhaps incrementing a deferral count */
}
else
{
  /* okay to access shared memory */
}
```

Listing Six

```
MY_STATUS my_status;
/* set inuse flag */
status_ptr->inuse = 1;
/* copy shared memory to local structure */
memcpy(&my_status, status_ptr, sizeof(MY_STATUS));
/* clear inuse flag */
status_ptr->inuse = 0;
/* refer to local struct from now on */
if (my_status.stat1 == 1)
{
  ...
}
```

Listing Seven

```
if (0 != status_ptr->inuse)
{
  /* defer status write, perhaps incrementing deferral count */
}
else
{
  /* okay to write status */
}
```

Listing Eight

```
#define ERROR_NUM 64  /* max number of error strings to be queued */
#define ERROR_LEN 256  /* max string length for an error */
char error[ERROR_NUM][ERROR_LEN];
```

Listing Nine

```
#define ERROR_NUM 64  /* max number of error strings to be queued */
#define ERROR_LEN 256  /* max string length for an error */
typedef struct
{
  unsigned char inuse;
/* flag signifying Linux accessing */
  char error[ERROR_NUM][ERROR_LEN]; /* the errors themselves */
  int start;
/* index of oldest error */
  int end;
/* index of newest error */
  int num;
/* number of items */
} MY_ERROR;
```

Listing Ten

```
/* initialize ring buffer; done once, perhaps in init_module() */
int error_init(MY_ERROR *errlog)
{
 errlog->inuse = 0;
 errlog->start = 0;
 errlog->end = 0;
 errlog->num = 0;
 return 0;
}
/* queue an error at the end */
int error_put(MY_ERROR *errlog, const char *error)
{
  if (errlog->num == ERROR_NUM)
    {
      /* full */
      return -1;
    }
  strncpy(errlog->error[errlog->end], error, ERROR_LEN);
  errlog->end = (errlog->end + 1) % ERROR_NUM;
  errlog->num++;
  return 0;
}
/* dequeue the error off the front */
int error_get(MY_ERROR *errlog, char *error)
{
  if (errlog->num == 0)
    {
      /* empty */
      return -1;
    }
  strncpy(error, errlog->error[errlog->start], ERROR_LEN);
  errlog->start = (errlog->start + 1) % ERROR_NUM;
  errlog->num--;
  return 0;
}
```

Listing Eleven

```
char error[ERROR_LEN];  /* place to copy error */
/* set in-use flag in shared memory */
errlog->inuse = 1;
/* copy error out */
if (0 != error_get(errlog, error))
{
  /* empty */
}
else
{
  /* handle it */
}
/* clear in-use flag in shared memory */
errlog->inuse = 0;
```

Listing Twelve

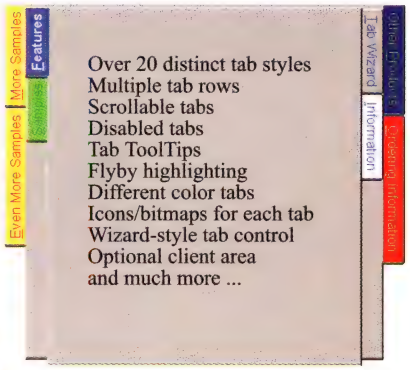
```
char error[ERROR_LEN];  /* place to compose error */
/* check for in-use */
if (0 != errlog->inuse)
{
  /* defer writing, perhaps incrementing deferral count */
}
else
{
  /* compose it */
  strcpy(error, "your error here");
  if (0 != error_put(errlog, error))
  {
    /* full */
  }
}
```

DDJ

softel vdm
NOT THE INDUSTRY STANDARD

Putting You In Control...

This tab control is available as SftTabs/DLL (a DLL-based control for use with Visual C++ and Borland C++) or as SftTabs/OCX (for use with Visual Basic).



Over 20 distinct tab styles
Multiple tab rows
Scrollable tabs
Disabled tabs
Tab ToolTips
Flyby highlighting
Different color tabs
Icons/bitmaps for each tab
Wizard-style tab control
Optional client area
and much more ...

Call today for a free catalog or demo!

Product evaluations available at www.softelvdm.com. Visit us today to find out about our Windows control products.

Softel vdm, Inc.
318 Tamiami Trail, Suite 14
Punta Gorda, FL 33950

(941) 505-8600
Fax (941) 505-8555
www.softelvdm.com

It all comes down to questions. Questions that challenge your expertise about Microsoft products. Question yourself – are you ready? Be absolutely sure. Once you've completed Spike and the gang's two step program, you'll pass with flying colors or get your money back.* Guaranteed.*

- Most Realistic MCSE and MCSD Simulations Available
- Detailed Answers and Explanations
- NEW! TranscenderFlash products
- Money Back If You Don't Pass Guarantee*
- From $129 - $179
- Add $4 U.S. Shipping and Handling, $25 Outside U.S.

Transcender® Corporation

Transcender. America's #1 Exam Preparation Software.

To order, call Howard @ (615) 726-8779 or fax (615) 726-8884; 242 Louise Ave.; Nashville, TN 37203.

www.transcender.com

© 1999 Transcender Corp. All Rights Reserved. Microsoft is a registered trademark of Microsoft Corporation. Multi-user licenses are available. *Call or see our Web site for details.

Inside Windows NT System Data

Exploring a mysterious function

Sven B. Schreiber

If you have ever written debugging software for Windows NT, you know it doesn't take long before you can get stuck. Things as simple as enumerating processes or querying system performance are difficult due to the lack of detailed documentation. In an interesting *Microsoft Systems Journal* (*MSJ*) article series, Matt Pietrek presented two methods for getting at some internal system data. The first one involves querying the so-called Performance Counters stored in the registry (see "Under The Hood," *MSJ*, March/April, 1996). The other method uses a special-purpose system module named psapi.dll ("Under The Hood," *MSJ*, August 1996). Both methods work well, but give just a tiny glimpse of the wealth of data used internally by Windows NT.

NtQuerySystemInformation()

While trying to figure out how the kill.exe utility (from the NT Resource Kit) works, I discovered it uses the mysterious *NtQuerySystemInformation()* function exported by ntdll.dll to get a list of processes currently running. With this list, kill.exe is able to force termination of a process by name, simply by looking up the name in the process list and retrieving the associated process ID, which is stored in this list, too. This looked interesting enough to warrant a closer look. To my surprise, that *NtQuerySystemInformation()* is apparently the general front door from user-mode code into internal kernel-mode data structures.

One of the things that puzzled me most was the complete lack of documentation on this function. Even the NT Device Driver Kit (DDK) header files ntddk.h and ntdef.h, that usually provide answers to many questions about NT internals, are amazingly ignorant about any aspects of *NtQuerySystemInformation()*. No API prototype, no info class definitions—just

a single mention of the name in the comments to the definition of the structure CONFIGURATION_INFORMATION. Searching for "NtQuerySystemInformation" in the DDK help yields no hits at all. That's about as undocumented as a system API ever could be! In this article, I'll reveal what kind of data this API gives you access to, and provide a DLL and demo program that lets you display most of that data on the screen (both available electronically; see "Resource Center," page 5).

It's simple to find out what parameters have to be passed to this function—you simply check how the NT core modules make use of it. Example 1 is the prototype of *NtQuerySystemInformation()*, as well as that of its counterpart, *NtSetSystemInformation()*. Actually, they look similar to other (documented) NT info query APIs, such as *NtQueryInformationProcess()* or *ZwQueryValueKey()*. All of them use an *info* class parameter, which is just an index number that selects the kind of information to be returned. *pData* points to a data buffer allocated by the caller, the size of which is indicated by *dSize*. On return, *pdSize* contains the number of bytes actually copied to the buffer. Like its friends, *NtQuerySystemInformation()* returns an NTSTATUS value. If *STATUS_INFO_LENGTH_MISMATCH (0xC0000004)* is returned, the buffer was too small to hold the data.

A Simple System Info Spy

To find out more about the data supplied by *NtQuerySystemInformation()*, I wrote the ntqsi.exe utility (available electronically), a simple console-mode application that outputs a hex dump of the data returned by *NtQuerySystemInformation()*. The problem with this function is that it returns several types of information, dependent on the specified info class. I've observed the following variations:

- A fixed-size data block (usually a scalar value or some structure).
- A variable-size list of fixed-size items.
- A variable-size list of variable-size items.
- A list of lists.

As you see, almost any kind of data you can think of is included. This makes it difficult to write a general-purpose spy utility that properly dumps out any data returned by *NtQuerySystemInformation()* without knowing in advance what this data is like. Unfortunately, the function does not tell

Sven is a developer in Herzogenaurach, Germany, and be contacted at sbs@orgon.com.

What Good Is An Object Tool Without An Object Database?

Attracting and retaining customers requires applications that are not only entertaining and informative, but also easy to use.

Today's business information is made up of not only text and numbers, but also images, motion, and sound.

Jasmine's object technology is designed for building enticing systems that you get to market faster, develop more rapidly, and gain productivity through reuse and integration of code, data and applications.

Only a pure object database like Jasmine inherently understands how to store and process complex data. Complex queries, such as fare comparisons for specific destinations and dates, are easily implemented with Jasmine.

Jasmine Works With Your Favorite Tool

- Sun:
  - Java Studio
  - Java WorkShop
- Microsoft:
  - Visual InterDev
  - Visual C++
  - FrontPage
  - Visual J++
  - VisualBasic
  - Visual Studio
- Symantec Visual Café
- Inprise JBuilder
- Bluestone Software
- Centura
- SilverStream
- SuperNova
- Dataviews Corporation
- Trellix

Just think how much more powerful your object tool would be with an object database like Jasmine™. With Jasmine's pure, object-oriented architecture, objects defined in your application tool (C++, Java, and others) are the same as the objects managed by Jasmine.

INFORMATION IN HARMONY™

You can use native language bindings, rather than learning and coding new non-object-based data access protocols such as JDBC. And Jasmine supports large amounts of complex data so you can deliver true, next-generation applications.

With Jasmine, objects never have to be converted to relational formats, so they run much faster. And there's no need to choose between object models—Jasmine supports both COM and Java. It even lets you use different tools at the same time. Jasmine's object infrastructure also integrates external data and applications, making them accessible from any application development tool.

That includes the object tool you're using right now. So call **1-888-7JASMINE** or visit **www.cai.com/ads/jasmine/dev** for your FREE, full-function Jasmine Developer Edition CD. And turn your object tool into an instant advantage.

COMPUTER® ASSOCIATES
Software superior by design.

Jasmine™

©1998 Computer Associates International, Inc., Islandia, NY 11788-7000.
All product names referenced herein are trademarks of their respective companies.

(continued from page 40)
you how large your buffer should be if it returns with STATUS_INFO_LENGTH_MISMATCH. (Closer examination of the implementation in ntoskrnl.exe reveals that the kernel also doesn't know beforehand how much data will be returned—it usually bails out as soon as it's about to write beyond the end of the buffer and discards any waiting data.)

Microsoft programs that have to deal with lists of the variable-size type (like, for instance, process lists) use a trial-and-error strategy to find out the correct buffer size: They start off with a reasonable initial size (32 or 64 KB, for instance), and increase the size by a fixed increment if *NtQuerySystemInformation()* fails with a length mismatch status. In the ntlib.c/ntinfo.c demo programs (available online), I use multiples of 64 KB, which seem to be a good choice based on my empirical tests.

The situation is further complicated by some info classes accepting larger buffers than required, while others insist on receiving a properly sized buffer. Some might report success although the buffer is actually too small, and still others fail to report the number of returned bytes (for example, *set *pdSize = 0*). Therefore, ntqsi.c uses a somewhat clumsy, but successful, approach to determine the correct size:

- It starts with an initial size that is chosen large enough to hold even very large lists (256 KB) and calls *NtQuerySystemInformation()* repeatedly, decrementing the size by 1 until the buffer size is 0 or a status code other than STATUS_INFO_LENGTH_MISMATCH is detected.
- If the status code is STATUS_SUCCESS and the reported size is nonzero, this size is assumed to be the correct buffer size.
- If the status code signaled success and the reported size is 0, the program starts with a 1-byte buffer and keeps on growing it while calling *NtQuerySystemInformation()* until a status other than STATUS_INFO_LENGTH_MISMATCH is returned. If it's STATUS_SUCCESS, the current buffer size is the desired value.

All info classes returning variable-size data succeed after the first call to *NtQuerySystemInformation()* and set the result size properly. Listing One is a sample implementation. Many info classes returning fixed-size data blocks succeed as soon as the buffer size is decremented to the correct size, and set the result size to this value. The second pass is only needed for some odd info classes that behave unlike the standard.

Of course, this heuristic is time consuming for info classes that return just a couple of bytes and require an exact size specification. However, the latency is in the range of just a few seconds on Pentium machines. (On a 400-MHz Pentium II, it takes 16 seconds to dump the entire currently valid info class range of 0 through 45.) Anyway, this is the only general approach I know of that returns correct results for virtually all info classes defined for NT 4.0 SP5. Of course, you can optimize the algorithm by using *a priori* knowledge about the info classes, but I wanted to create a program that's expected to work reliably even on future versions of NT without change.

The general command-line format of ntqsi.exe is: *ntqsi [[+p | -p] [/<error limit>] [<class> | [<class#1>]-[<class#n>]]].*

Pointer Relocation

In the most simple case, you can enter, say, *ntqsi 18*, and get a hex dump containing some familiar looking Unicode text. Obviously, this is some information about your page-file configuration. Example 2 is a sample printout. Now try the same command with *+p* inserted before the number, and you should get something like the hex dump in Example 3. There's not much difference, except for the 4 bytes at offset 0x14 (marked by "=" characters in between). As you'll see, this is the "Buffer" member of a UNICODE_STRING structure (that is, a pointer to a Unicode string) holding the path of a page file.

If *NtQuerySystemInformation()* returns pointers inside the result data, they always point to addresses inside the caller's buffer. This ensures that all pointers remain valid as long as the caller is working with the returned data, and that deallocating the data buffer also frees all data blocks referenced by those pointers. ntqsi.exe is clever enough to detect pointers inside the returned data stream and marks them with "=" characters between the 4 bytes forming the pointer (see Examples 2 and 3, offset 0x14). Moreover, it subtracts the address of its internal buffer from the pointer, thus normalizing it to become a 0-based offset into the data. If you read out the value at offset 0x14 in Example 3, you get 0x00000018, and that's exactly where you'll find a sequence of Unicode characters making up the file path "D:\pagefile.sys."

Of course, ntqsi.exe lets you specify ranges of info classes, like *ntqsi 0–45*, or sets like *ntqsi 1 2 5 3*. In a range specification, you can omit the starting or ending class, or even both. Omitting the first part of the pair defaults to 0, omitting the second part means "go on forever." (There is an upper limit, imposed by the number of values a DWORD variable can take.) Because the latter case might produce tons of garbage as soon as the last info class is transcended, it's wise to use the */<error limit>* option here. For instance, using *ntqsi /7 37-* will dump all info classes starting at 37, and stop after seven consecutive errors have occurred.

ntqsi.exe attempts to find out which members of a structure are actually used by the system. Therefore, it fetches the data twice after presetting the buffer with certain bit patterns. On the first run, it uses 0x55 for every byte, and 0xAA on the second run. Every byte position that contains 0x55 after the first and 0xAA after the second call is considered to be unused. Those bytes are marked by a double dash in the hex output.

The NT Kernel Debugger

Although it's interesting to study hex dumps of *NtQuerySystemInformation()*, it would be more exciting to know what all those bytes are about. That's really difficult, and it took me several weeks to compile enough information to get a basic understanding of the returned data. If you ever try to gain information about undocumented NT interfaces and don't happen to have access to the source code, there's probably no way out other than using the

```
NTSTATUS NTAPI
NtQuerySystemInformation (SYSTEMINFOCLASS sic,
                          PVOID           pData,
                          DWORD           dSize,
                          PDWORD          pdSize);
NTSTATUS NTAPI
NtSetSystemInformation (SYSTEMINFOCLASS sic,
                        PVOID           pData,
                        DWORD           dSize);
```

Example 1: *The prototypes of* NtQuerySystemInformation() *and* NtSetSystemInformation().

```
NtQuerySystemInformation (18) -> 64 bytes at 0x007B0020

Offset¦ 00 01 02 03 04 05 06 07 : 08 09 0A 0B 0C 0D 0E 0F ¦ 01234567 : 89ABCDEF
------¦-----------------------:-----------------------¦----------:---------
00000 ¦ 00 00 00 00 00 8B 00 00 : 66 0B 00 00 66 0B 00 00 ¦ ....... : f...f...
00010 ¦ 26 00 28 00 38=00=7B=00 : 5C 00 3F 00 3F 00 5C 00 ¦ &.(.8.{. : \.?.?.\.
00020 ¦ 44 00 3A 00 5C 00 70 00 : 61 00 67 00 65 00 66 00 ¦ D.:.\.p. : a.g.e.f.
00030 ¦ 69 00 6C 00 65 00 2E 00 : 73 00 79 00 73 00 00 00 ¦ i.l.e... : s.y.s...
```

Example 2: *The results of the command of* ntqsi 18.

Sleep Well – NuMega™ Error Detection Tools are on the Job

If you're caught in a seemingly hopeless cycle of shorter deadlines, limited resources and increased demand for applications, the last thing you want to do is spend countless hours tracking down bugs in your code. Working another late night is not the answer. NuMega™ error detection software from Compuware® is.

If you're like most developers, you're probably using traditional manual debugging methods. So, you know about half of your development time is spent debugging rather than coding. It doesn't have to be this way...

NuMega error detection tools can save you time and shorten your development cycle by automating the error detection and debugging process. Whether you're working in Visual C++™, Visual Basic® or Java™, NuMega tools continuously check for errors as you create your application, alert you immediately to software errors and recommend solutions. Early detection means quicker fixes, more reliable code and earlier nights for you.

Regain your productivity with NuMega tools. Write better, faster and more reliable code. And deliver applications on time. Your customers will be happy, and *you'll* get some well-deserved sleep.

So, *now* what are you doing tonight?

Accelerating Software Development with
NuMega Error Detection Tools from Compuware:

For Visual C++ – NuMega BoundsChecker® Visual C++ Edition
For Visual Basic – NuMega SmartCheck™
For Java – NuMega JCheck™

Ask about NuMega DevPartner™ Studio – A powerful suite of *error detection, performance tuning* and *code coverage solutions*!

Best Value!

Call for a FREE 14-day license, or visit us on the Web for a quick product tour.

1-800-4-NUMEGA
www.compuware.com/numega

COMPUWARE®
People and software for business applications™

30-Day Money Back Guarantee

Copyright © 1999 Compuware Corporation. All rights reserved. All trademarks and registered trademarks are the property of their respective owners.
Java and all Java-based marks are trademarks of Sun Microsystems, Inc. in the United States and other countries.

(continued from page 42)
NT Kernel Debugger (KD), along with the symbol files on your NT setup CD (look for \support\debug\i386\symbols).

The KD comes in two flavors: the GUI version windbg.exe shipping with the Platform SDK, and the console-mode version i386kd.exe (or alphakd,exe, mipskd.exe, or ppckd.exe, if you happen to use a platform other than Intel x86) included on the NT setup CD (\support\debug\<platform>). I recommend using i386kd.exe, because it features a much more powerful instruction set than windbg.exe. In *Inside Windows NT*, Second Edition (Microsoft Press, 1998), David A. Solomon gives detailed instructions on how to set up i386kd.exe and the symbol files, using the information from a forced crash dump system image. The basic steps are:

```
NtQuerySystemInformation (18) -> 64 bytes

Offset¦ 00 01 02 03 04 05 06 07 : 08 09 0A 0B 0C 0D 0E 0F ¦ 01234567 : 89ABCDEF
------¦-----------------------:-----------------------¦----------:---------
00000 ¦ 00 00 00 00 00 8B 00 00 : 66 0B 00 00 66 0B 00 00 ¦ ....... : f...f...
00010 ¦ 26 00 28 00 18=00=00=00 : 5C 00 3F 00 3F 00 5C 00 ¦ &.(..... : \.?.?.\.
00020 ¦ 44 00 3A 00 5C 00 70 00 : 61 00 67 00 65 00 66 00 ¦ D.:.\.p. : a.g.e.f.
00030 ¦ 69 00 6C 00 65 00 2E 00 : 73 00 79 00 73 00 00 00 ¦ i.l.e... : s.y.s...
```

Example 3: *The results of the command of* "ntqsi +p 18."

1. Enable creation of a crash dump image in the Control Panel (System applet, "Startup/Shutdown" tab). Also, be sure to increase your page file size (twice the amount of physical memory is a good value) so you won't run into a virtual memory problem later.
2. Force a crash dump by killing the "Client Server Runtime Process" csrss.exe, which is the module containing the Win32 subsystem. (You need kill.exe from the NT Resource Kit or an equivalent utility that is able to grab enough privileges to terminate a system service.)
3. Reboot so the crash dump image is copied from the page file to the file specified in the Control Panel (%systemroot%\memory.dmp by default). Note that your system will run quite low on virtual memory after doing that, so you should reboot once more. (That's why I recommended increasing the page file size in the first step.)
4. Set the _NT_SYMBOL_PATH environment variable to the base directory of the symbol files. This directory contains a couple of subdirectories, like "dll," "drv," "exe," and the like.
5. Issue the command *i386kd -z <crash dump image>*, where "<crash dump image>" is the full path of the crash dump file created before.

Having done this, you can explore a snapshot of the system that was valid at the time you were killing csrss.exe. You can't access some system data structures in user-mode memory, like the process and thread environment blocks. You need a live system to explore those. Now, at the kd> prompt, you can enter the debugger commands of your choice.

There are both built-in commands and external bang commands prefixed by a "!" character, which are processed by an extension DLL. i386kd.exe uses the extension DLL kdextx86.dll by default (kdextalp.dll, kdextmip.dll, and kdextppc on the other platforms). Those DLLs export a couple of APIs that are named like the corresponding bang commands they handle. Enter *!help* at the kd> prompt to get a list of bang commands supported by the extension DLL currently in use.

Retrieving Internal NT Symbols

Probably the most useful kernel debugger command is *ln* (list nearest symbol). You can specify either a hex address or a symbol name after the *ln* token, and you'll get the name and address of the symbol whose address is below the specified one and has the least distance from it. It also displays the address and name of the next symbol, so you get an idea of how large the data addressed by the symbol actually is. (Some symbols are

obviously missing, so the distance between two symbols can be misleading.)

When you're exploring NT internals, you usually don't have more than a disassembly listing at hand, which will not contain more symbolic information than is exported from the NT system modules. The *ln* debugger command helps you convert opaque binary addresses into meaningful symbols that might suggest what this address is used for. Specifically, you can look up the names of all internal functions used by the code you're currently examining. (If you're still looking for a good Win32 disassembler, you might want to contact Jean-Louis Seigne at jls@chez.com who has written a great one.)

Another useful KD command, *x nt!**, yields a complete list of symbols defined inside the NT kernel. And finally, the commands *db* (dump BYTEs), *dw* (dump WORDs), and *dd* (dump DWORDs) let you see hex dumps of memory regions in the appropriate number format.

Although the APIs *NtQuerySystemInformation()* and *NtSetSystemInformation()* are exported by ntdll.dll, the actual implementation is located in a different module: ntoskrnl.exe. As Example 4 shows, the implementation of those APIs inside ntdll.dll is trivial. The secret of using INT 2Eh is that it serves as a gate from the processor's user mode to kernel mode, so the handlers can execute with enough privileges to access kernel-mode code and data. (On other CPU platforms, different but equivalent means are provided to achieve this effect.)

On the other side of the INT 2Eh gate, the NT operating system kernel (ntoskrnl.exe) is waiting. It uses the index value in register EAX to look up an entry in the kernel's service descriptor table. This entry contains the information that's needed to copy the required number of bytes from the caller's stack (addressed by EDX) to a system stack, and to call the associated handler. Under Windows NT 4.0 SP3, the handlers of *NtQuerySystemInformation()* and *NtSetSystemInformation()* happen to be located in the .PAGE section of ntoskrnl.exe at 0x801558c2 and 0x80156a48, respectively. You can use the kernel debugger's *ln* command to verify that.

System Information Classes

In Windows NT Version 4.0 SP5, *NtQuerySystemInformation()* and *NtSetSystemInformation()* recognize info class values in the range of 0 up to 45. However, not all info classes will return a success status. The info classes can be put in one or more categories of the following:

- Read-only info classes (not to be used with *NtSetSystemInformation()*).
- Write-only info classes (not to be used with *NtQuerySystemInformation()*).
- Info classes only available in the "Checked Build" of Windows NT.
- Invalid info classes (either defunct or superseded by other info classes or system APIs).

To keep things simple, I deal almost exclusively with the first category. Table 1 lists all info classes I've been able to identify.

A Custom Header File for ntdll.dll

To save you from the hassle of directly interfacing to ntdll.dll, I've designed the

```
NtQuerySystemInformation:
77f67d2c b87c000000      mov     eax,0x7c
77f67d31 8d542404        lea     edx,[esp+0x4]
77f67d35 cd2e            int     2e
77f67d37 c21000          ret     0x10

NtSetSystemInformation:
77f68034 b8ae000000      mov     eax,0xae
77f68039 8d542404        lea     edx,[esp+0x4]
77f6803d cd2e            int     2e
77f6803f c20c00          ret     0xc
```

Example 4: *Sample API gates inside ntdll.dll.*

library ntlib.dll, which contains high-level APIs to load and parse the more complex data sets returned by *NtQuerySystemInformation()*, as well as several utility functions for device-driver management, memory size, and time conversion. It also features a powerful command-line parser with full-blown option management and command help support. The associated utility program ntinfo.exe demonstrates how to make use of it. The source code of both programs is available electronically.

The main problem with ntdll.dll is that neither the Win32 Platform SDK nor the NT DDK provide appropriate header files that can be used in user mode applications. The main DDK header file ntddk.h is designed for kernel-mode drivers, which don't rely on the windows.h file required for Win32 applications. Hence, Microsoft didn't care to make them compatible. If you ever try to include both of them in a single project, the compiler will emit tons of errors and warnings. To remedy this, I'm using the same approach as Matt Pietrek in his *MSJ* articles: Create my own header file that contains just enough definitions to be able to interface to ntdll.dll, while not sacrificing windows.h compatibility. Now, what name would be most appropriate for a file like that? Yes, of course: NtDll.h.

#	Name	Comments
0	SystemBasicInformation	Basic system parameters, such as system clock frequency, memory configuration, and number of processors.
1	SystemProcessorInformation	CPU architecture, family, revision, and features.
2	SystemPerformanceInformation	Various performance counters, like I/O operations, transfer loads, and memory/cache usage.
3	SystemTimeInformation	Startup/current time, time zone information.
4	SystemPathInformation	(Not implemented.)
5	SystemProcessInformation	List of all running processes/threads, with detailed property lists per item.
6	SystemCallInformation	Usage of kernel and Win32k system calls. (Checked build only.)
7	SystemConfigurationInformation	Number of peripheral devices attached to the system (hard/floppy disks, CD-ROM and tape drives, serial and parallel ports).
8	SystemProcessorCounters	Elapsed processor time spent in various modes.
9	SystemGlobalFlag	Status of the NT Global Flag (ntoskrnl.NtGlobalFlag).
10	SystemInfo10	(Not implemented.)
11	SystemModuleInformation	List of loaded drivers, with detailed property lists per item.
12	SystemLockInformation	List of all ERESOURCE locks currently in use, with detailed usage statistics.
13	SystemInfo13	(Not implemented.)
14	SystemPagedPoolInformation	(Checked build only.)
15	SystemNonPagedPoolInformation	(Checked build only.)
16	SystemHandleInformation	List of all open handles in all processes.
17	SystemObjectInformation	Supposed to return a list of all objects currently in use, grouped by object type. However, actually returns the LPC port object list only.
18	SystemPageFileInformation	List of all active page files.
19	SystemInstemulInformation	Usage of the V86 instruction emulation handlers.
20	SystemInfo20	(Invalid info class.)
21	SystemCacheInformation	Cache usage counters.
22	SystemPoolTagInformation	Allocation/deallocation statistics for the paged and nonpaged pools, grouped by pool tags.
23	SystemInfo23	CPU-specific information. (Multiprocessor systems only. No details available.)
24	SystemDpcInformation	Various DPC parameters.
25	SystemInfo25	(Checked build only.)
26	SystemLoadDriver	Load a kernel mode driver without initializing it. (NtSetSystemInformation() in kernel mode only.)
27	SystemUnloadDriver	Unload a kernel mode driver. (NtSetSystemInformation() only in kernel mode.)
28	SystemTimeAdjustmentInformation	Parameters for periodic time-of-day clock adjustment (see Win32 Get/SetSystemTimeAdjustment()).
29	SystemInfo29	(Checked build only.)
30	SystemInfo30	(Checked build only.)
31	SystemInfo31	(Checked build only.)
32	SystemCrashDumpInformation	Handle to memory section containing system image to be written to the crash dump file.
33	SystemInfo33	(No details available.)
34	SystemCrashDumpStateInformation	Flag indicating if crash dumps are enabled.
35	SystemDebuggerInformation	Flags indicating if a kernel debugger is present and enabled.
36	SystemThreadSwitchInformation	Find, idle, and preemption statistics across all processors.
37	SystemRegistryQuotaInformation	Registry size and usage.
38	SystemAddDriver	Load and initialize a kernel-mode driver. Callable from user mode if the SeLoadDriverPrivilege is held. (NtSetSystemInformation() only.)
39	SystemPrioritySeparationInformation	Set the value of the internal system parameter dPsPrioritySeparation. (NtSetSystemInformation() only.)
40	SystemInfo40	(Not implemented.)
41	SystemInfo41	(Not implemented.)
42	SystemInfo42	(Invalid info class.)
43	SystemInfo43	(Invalid info class.)
44	SystemTimeZoneInformation	Time-zone-specific parameters related to daylight savings time.
45	SystemLookasideInformation	List of all lookaside lists currently used, with allocation, deallocation, and no-miss statistics.

Table 1: *Available system information classes.*

NtDll.h is designed to be modular. It defines macros, constants, and structures that are used in almost any part of the NT system. Near the end of the file, it yanks in the files NtObj.h, NtTeb.h, and NtSys.h. Only the latter is included in the online package and contains all definitions related to *NtQuerySystemInformation()* and *NtSetSystemInformation()*. The others are just stubs. The real things contain so much undocumented stuff about NT objects in general and thread/process objects in particular, that I decided to hold them back for a separate article on NT objects, hopefully to be available soon.

Displaying Structured NT System Information

The general command-line format of ntinfo.exe is: *ntinfo [<option> [[<parameter>]]]*. A command line may contain several options, which consist of a prefix character ("-" or "/") and a mnemonic name, such as *-pagefile* or *-process*. The option name can be abbreviated, provided that the remaining characters still identify it unambiguously. Hence, *ntinfo -p* is invalid, because it matches both *-pagefile* and *-process*. However, *ntinfo -pa* and *ntinfo -pr* are okay. Please note that ntinfo.exe cares to list all matching options in the case of an ambiguous specification, using the command-line parser included in ntlib.dll.

Some options don't expect additional parameters, while others can take one or more optional parameters; see Table 2 for a summary of options. For instance, the *-pagefile* option takes no parameters and always lists all installed page files, while the output of the *-process* option can be filtered by specifying one or more process IDs.

Some options display just a subset of the available information by default, and you have to specify "*" or "all" to get the complete data. The *-object* option belongs to this category. Still others accept a *this*

There's no need to panic at release time.

Would you believe that Insure++® can automatically find your errors at the source code level?

"Insure++ has saved me dozens of hours of work. It has discovered and reported problems which no other tool could detect. If I had not had access to Insure++, I would still be debugging that same project a year later." -- ***Douglas Scott*** ***Silicon Graphics, Inc.***

Insure++® is an automatic runtime error detection tool for C and C++. Forget what you know about error detection tools, because Insure++ is different: it automatically tests your code at the source level. The result for you is more accurate results with less work.

If you like to get technical, you'll want to know that Insure++'s unique bug-finding ability comes from two powerful, patented technologies. These technologies help Insure++ to build a database of your program's elements and then check values against it at runtime. But all you really need to understand is that on every run, Insure++ reports the exact location and complete explanation of each error, helping you to make speedy repairs.

Best of all, Insure++ is tailored to meet your needs at all stages of the development process. When you want to perform faster debugging or to get a general idea of where bugs exist, you can use Insure++'s intermediate code instrumentation mode to scan your program. You can then zero in on a problematic area and run Insure++ again in its default mode to receive more detailed error information. You'll hunt down even the most elusive bugs in record time.

If you develop on Windows®, Insure++ will integrate directly into Microsoft® Developer Studio®, and you can start finding bugs by clicking an icon. Insure++ also caters to all UNIX platforms, on which it can be installed and run with a few simple commands.

Two add-ons help make Insure++ a comprehensive testing package. The Inuse® add-on graphically displays memory usage, helping you find leaks and increase your understanding of memory. The TCA® add-on reports on the completeness and frequency of your testing, so you don't have to worry about where bugs may still be hiding in your code.

If you would like to find more errors in less time and release every project with confidence, download a demo of Insure++ today at www.parasoft.com/story.htm, or call (888)305-0041 for more information.

ParaSoft®

www.parasoft.com/story.htm

ParaSoft is a registered trademark and Insure++, Inuse and TCA are registered trademarks of ParaSoft Corporation. Microsoft Developer Studio and Windows are registered trademarks of Microsoft in the US and other countries.

www.LearningPatterns.com
Your Global Source for Java Learning
888-298-2677
See Us in Booth 318 at SD East99

Option	Parameters	Comment
-basic	None	Basic system and processor configuration.
-cpu	None	CPU identification and usage.
-time	None	Time-of-day, time zone, and system time information.
-memory	None	Memory usage.
-cache	None	Cache usage and performance.
-pagefile	None	Page file configuration.
-lock	[*|all]	List of ERESOURCE locks.
	No parameters	All locks where at least one counter is nonzero; */all: all locks, regardless of counter values.
-module	[*|all] { [this|<ID>] }	List of modules.
	No parameters or v/all	All kernel-mode drivers; this: user-mode modules loaded by ntinfo.exe; <ID1> ... <IDn>: modules matching one of the specified IDs.
-handle	[*|all] { [this|<PID>] }	List of open handles.
	No parameters or */all:	All open handles held by all processes; this: all handles held by ntinfo.exe; <PID1> ... <PIDn>: all handles held by the specified processes.
-object	[*|all] { [this|<PID>] }	List of LPC port objects.
	No parameters	All named objects referenced by all processes; */all: all objects — named and unnamed — referenced by all processes; <PID1> ... <PIDn>: all objects referenced by the specified processes.
-process	[*|all] { [this|<PID>] }	List of processes.
	No parameters or */all	All processes in the system; this: process info about ntinfo.exe; <PID1> ... <PIDn>: info about the specified processes.
-thread	[*|all] { [this|<PID>|<TID>] }	List of threads.
	No parameters or */all	All threads in the system; this: list of threads started by ntinfo.exe; <PID1> ... <PIDn>: threads started by the specified processes; <TID1> ... <TIDn>: info about the specified threads.
-lookaside	[*|all] { [<tag pattern>] }	List of lookaside lists.
	No parameters or */all:	All lookaside lists in the system; <tag pattern1> ... <tag patternN>: all lookaside lists with a tag that matches at least one of the specified patterns (case sensitive).
-help	None	Ntinfo command help.
-?	None	Ntinfo command help.

Table 2: *Options available from ntinfo.exe.*

(continued from page 46)
parameter that defaults to the current process, and which is, of course, ntinfo.exe itself. For example, *ntinfo -handle this* lists all handles currently held by ntinfo.exe. You can also specify a negative filter: If "*" or "all" is specified with "this" or an individual ID or pattern, it means that the result should contain all items except those specified in addition to "*" or "all."

Object Type ID	Name
1	Type Object
2	Directory
3	Symbolic Link
4	Token
5	Process
6	Thread
7	Event
8	Event Pair
9	Mutant
10	Semaphore
11	Timer
12	Profile
13	Window Station
14	Desktop
15	Section
16	Key
17	Port
18	Adapter
19	Controller
20	Device
21	Driver
22	I/O Completion Port
23	File

Table 3: *Object type IDs used by the* SystemHandleInformation *class.*

It's remarkable that this undocumented system interface hasn't changed very much in Windows 2000. The latest Beta available at this writing (Release Candidate 1) adds two more info classes, and introduces minor changes in a few data structures. For example, the *ntinfo* option *"-thread"* displays garbage, *"-lookaside"* fails on some entries, and *"-object"* might throw an exception while trying to display the new *"WaitablePort"* object list. However, the remaining options appear to work correctly.

Conclusion

The system information interface discussed here and exposed by the ntdll.dll API pair *NtQuerySystemInformation()* and *NtSetSystemInformation()* offers functions — querying system configuration items and process/thread or system module lists — no system utility developer can live without.

DDJ

Listing One

```
NTSTATUS WINAPI QuerySystemInformation (SYSTEMINFOCLASS sic,
                                        PPVOID          ppData,
                                        PDWORD          pdData)
    {
    PVOID    pData;
    DWORD    dData, n;
    NTSTATUS ns = STATUS_INVALID_PARAMETER;
    dData = 0;
    if (ppData != NULL)
        {
        n = 0;
        while ((pData = LocalAlloc (LMEM_FIXED, dData += 0x10000))
               != NULL)
            {
            ns = NtQuerySystemInformation (sic, pData, dData, &n);
            if (ns != STATUS_SUCCESS) n = 0;
            if (ns != STATUS_INFO_LENGTH_MISMATCH) break;
            LocalFree (pData);
            }
        dData = n;
        if (pData != NULL)
            {
            if (ns != STATUS_SUCCESS)
                {
                LocalFree (pData);
                pData = NULL;
                dData = 0;
                }
            }
        else
            {
            ns = STATUS_NO_MEMORY;
            }
        *ppData = pData;
        }
    if (pdData != NULL) *pdData = dData;
    return ns;
    }
```

Listing Two

```
// 05: SystemProcessInformation
//     see ExpGetProcessInformation()
//     see also ExpCopyProcessInfo(), ExpCopyThreadInfo()
typedef struct _SYSTEM_THREAD
    {
    QWORD        qKernelTime;       // 100 nsec units
    QWORD        qUserTime;         // 100 nsec units
    QWORD        qCreateTime;       // relative to 01-01-1601
    DWORD        d18;
    PVOID        pStartAddress;
    CLIENT_ID    Cid;               // process/thread ids
    DWORD        dPriority;
    DWORD        dBasePriority;
    DWORD        dContextSwitches;
    DWORD        dThreadState;      // 2=running, 5=waiting
    KWAIT_REASON WaitReason;
    DWORD        dReserved01;
    }
    SYSTEM_THREAD;
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
typedef struct _SYSTEM_PROCESS_INFORMATION
    {
    DWORD          dNext;           // relative offset
    DWORD          dThreadCount;
    DWORD          dReserved01;
    DWORD          dReserved02;
    DWORD          dReserved03;
    DWORD          dReserved04;
    DWORD          dReserved05;
    DWORD          dReserved06;
    QWORD          qCreateTime;     // relative to 01-01-1601

    QWORD          qUserTime;       // 100 nsec units
    QWORD          qKernelTime;     // 100 nsec units
    UNICODE_STRING usName;
    KPRIORITY      BasePriority;
    DWORD          dUniqueProcessId;
    DWORD          dInheritedFromUniqueProcessId;
    DWORD          dHandleCount;
    DWORD          dReserved07;
    DWORD          dReserved08;
    VM_COUNTERS    VmCounters;
    DWORD          dCommitCharge;   // bytes
    SYSTEM_THREAD  ast [];
    }
    SYSTEM_PROCESS_INFORMATION;
```

Listing Three

```
void WINAPI DisplayProcesses (void)
    {
    NTL_TABLE                   nt;
    PSYSTEM_PROCESS_INFORMATION pspi;
    TIME_FIELDS                 tf;
    DWORD                       i;
    if (NtlTableProcess (&nt) == STATUS_SUCCESS)
        {
        printf (T("PID Org BP Th Hdls CommChrg WSetSize ")
                T("PFCount Start date and time Name\r\n"));
        pspi = NtlTableFirst (&nt);
        for (i = 0; i < nt.dCount; i++)
            {
            NtlTimeUnpack (&pspi->qCreateTime, &tf);
            printf (T("\r\n%3lu %3lu %2lu %2lu %4lu %8lu %8lu ")
                    T("%7lu %02u-%02u-%04u %02u:%02u:%02u ")
                    T("\"%ls\""),
                    pspi->dUniqueProcessId,
                    pspi->dInheritedFromUniqueProcessId,
                    pspi->BasePriority,
                    pspi->dThreadCount,
                    pspi->dHandleCount,
                    pspi->dCommitCharge,
                    pspi->VmCounters.WorkingSetSize,
                    pspi->VmCounters.PageFaultCount,
                    tf.Month, tf.Day,    tf.Year,
                    tf.Hour,  tf.Minute, tf.Second,
                    (pspi->usName.Buffer != NULL
                     ? pspi->usName.Buffer
                     : L"Idle"));
            pspi = NtlTableNext (&nt, pspi);
            }
        NtlTableUnload (&nt);

        }
    return;
    }
```

Listing Four

```
// 11: SystemModuleInformation
//     see ExpQueryModuleInformation
typedef struct _SYSTEM_MODULE
    {
    DWORD dReserved01;
    DWORD d04;
    PVOID pAddress;
    DWORD dSize;                // bytes
    DWORD dFlags;
    WORD  wId;                  // zero based
    WORD  wRank;                // 0 if not assigned
    WORD  w18;
    WORD  wNameOffset;
    BYTE  abName [MAXIMUM_FILENAME_LENGTH];
    }
    SYSTEM_MODULE;
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
typedef struct _SYSTEM_MODULE_INFORMATION
    {
    DWORD         dCount;
    SYSTEM_MODULE asm [];
    }
    SYSTEM_MODULE_INFORMATION;
```

Listing Five

```
// 16: SystemHandleInformation
//     see ExpGetHandleInformation ()
typedef struct _SYSTEM_HANDLE
    {
    DWORD       dIdProcess;
    BYTE        bObjectType;    // OB_TYPE_*
    BYTE        bFlags;         // bits 0..2 HANDLE_FLAG_*
    WORD        wValue;         // multiple of 4
    POBJECT     pObject;
    ACCESS_MASK GrantedAccess;
    }
    SYSTEM_HANDLE;
```

Listing Six

```
// 18: SystemPageFileInformation
//     see MmGetPageFileInformation()
typedef struct _SYSTEM_PAGE_FILE_INFORMATION
    {
    DWORD          dNext;       // relative offset
    DWORD          dTotal;      // pages
    DWORD          dInUse;      // pages
    DWORD          dPeak;       // pages
    UNICODE_STRING usName;
    }
    SYSTEM_PAGE_FILE_INFORMATION;
```

Listing Seven

```
// 26: SystemLoadDriver (set mode only)
//     see MmLoadSystemImage()
//     user mode: STATUS_PRIVILEGE_NOT_HELD returned
typedef struct _SYSTEM_LOAD_DRIVER
    {
    UNICODE_STRING usImageFile;     // input
    PVOID          pBaseAddress;    // output
    HANDLE         hSystemImage;    // output
    PVOID          pEntryPoint;     // output
    PVOID          pDirectoryEntry; // output
    }
    SYSTEM_LOAD_DRIVER;
// 27: SystemUnloadDriver (set mode only)
//     see MmUnloadSystemImage()
//     user mode: STATUS_PRIVILEGE_NOT_HELD returned
typedef struct _SYSTEM_UNLOAD_DRIVER
    {
    HANDLE hSystemImage;            // received via SystemLoadDriver
    }
    SYSTEM_UNLOAD_DRIVER;
// 38: SystemAddDriver (set mode only)
//     see MmLoadSystemImage(), MmUnloadSystemImage()
//     user mode: SeLoadDriverPrivilege required
typedef struct _SYSTEM_ADD_DRIVER
    {
    UNICODE_STRING usImageFile;
    }
    SYSTEM_ADD_DRIVER;
```

DDJ

Extending the PalmOS

Encapsulating functionality with shared libraries

Greg Winton

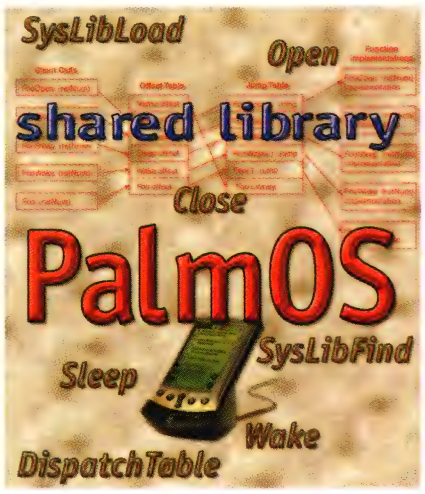
Packaging code into shared libraries provides many benefits. For one thing, libraries provide a mechanism for reusing your code among two or more applications, thereby reducing development time. For another, multiple applications can use the same library, reducing both the installed and in-memory footprint. The lifetime of a shared library is not strictly tied to that of its clients, so its state can be maintained across applications. Finally, architecturally, shared libraries reduce coupling between modules, which results in more stable and maintainable code.

Although there is limited support from the CodeWarrior tools, a handful of articles and white papers, and one example on the Palm web site, information on shared libraries for the Palm Computing Platform is still hard to come by. Consequently, in this article, I'll examine the Palm shared library model and present a shared library (available electronically; see "Resource Center," page 5) that you can use.

The Palm Shared Library Model

A shared library is a special type of database that provides services to clients; these clients may be either Palm applications or other shared libraries. These clients refer to the shared library by its reference number, and invoke functions published by the library.

From the client side, use of a shared library is straightforward. The client loads the shared library into memory using *SysLibLoad*. If the library has already been loaded, this function will fail—the client should then call *SysLibFind* to locate the loaded library. System libraries, such as *SerialMgr* and *IrMgr*, are always loaded, so clients of these libraries can skip the load step. Custom shared libraries are not initially loaded, and thus must be loaded by the client.



On success, *SysLibLoad* and *SysLibFind* both return a library reference number, which must be used for all subsequent calls to the shared library's functions. The convention is that the first function to be called must be the *Open* function. Then the client may call the rest of the library's API.

When the client is finished with the shared library, it calls the library's *Close* function. It may also need to unload the library using *SysLibRemove*. It is recommended that *Close* should return success if it can be unloaded, or an error if it still has active clients.

In conventional Palm development scenarios, functions are linked in as you would expect with any other platform. It is when you invoke functions defined in shared or system libraries that things get interesting.

If you look at any system header, you will notice that every function declaration is followed by the SYS_TRAP macro. The argument of this macro is the function trap identifier. This declaration syntax causes the function's arguments to be placed on the stack and then invokes the system trap dispatcher. This dispatcher maps the trap identifier to a specific function call.

The trap identifiers for system libraries are unique throughout the operating system. Shared library function trap identifiers are unique only within the library. As a shared library developer, you have to define the trap identifiers for your library, and implement the mechanism whereby traps are mapped to function calls.

The system provides three services for shared libraries:

- The system trap dispatcher dispatches calls by library reference number to the appropriate library. When a client invokes a function by a trap identifier, it generates an exception. The trap dispatcher handles this exception by putting the address of the appropriate function onto the stack. When the exception returns, program execution is transferred to this address.
- For each shared library open in the system, there is a corresponding entry in the system library table. This entry holds both the library's dispatch table and the memory allocated by the library for its global memory.
- The library reference number is assigned by the system when the library is loaded for the first time. The client retrieves this number either from *SysLibLoad*, which loads the library, or *SysLibFind* if the library is already loaded. This reference number must be the first parameter to every shared library call; the system trap dispatcher uses it to locate the correct library.

The shared library provides the mechanism by which trap identifiers are translated into function calls. There are several

Greg is a director of software development for Bachmann Software and a contributor to Palm Programming, *by Glenn Bachmann (Sams Publishing, Macmillan Computer Publishing, 1999). Greg can be contacted at greg@bachmannsoftware.com or at http://www.bachmannsoftware.com/.*

6 Ways to Higher Profitability with Electronic Licensing

- Avant! Corporation
- Avid Technology
- Bristol Technology
- British Telecom PLC
- Cadence Design Systems
- Cayenne Software
- Cisco Systems
- Cognos Inc.
- Compaq
- Computer Associates
- Compuware Corp.
- Digital Equipment
- Eastman Kodak
- EDS
- Ericsson
- ESRI
- Fujitsu
- GEC Plessey
- GTE
- Hewlett Packard
- Honeywell
- IBM
- Intel
- Landmark Graphics
- Logicon
- Lucent
- MacNeal Schwendler Corp.
- MapInfo
- Mathworks
- Mentor Graphics
- Net Perceptions Inc.
- Nortel
- Parametric Technology
- Pixar
- Platinum Technology
- Rational
- Schlumberger
- SDRC
- Silicon Graphics
- Sun Microsystems
- Synopsys
- Texas Instruments
- Unisys
- Unigraphics
- Xerox
- Xilinx
- Zuken

Over 1600 Software Publishers use FLEXlm

Make 40-50% more profit, increase account penetration, expand your market and slash lost revenues from unlicensed use with FLEX*lm*®(FLEXible License Manager®). Software industry analysts forecast that over 50% of software shipped will use electronic licenses within the next 5 years.

Over 1600 software publishers used FLEX*lm*'s patented technology in shipping over $10 billion of software in 1997.

GLOBE*trotter*'s FLEX*lm* works whether your software is distributed via the Internet, CD, direct sales, through integrators, or sold in a box at retail. Here's how FLEX*lm* increases profitability 6 ways by enforcing or monitoring user compliance with licensing terms:

1 – Slash unlicensed use. Lost revenue is not from thieves, but from your customers' unlicensed use. Customers will pay for software, but without electronic licensing customers don't know what licenses they're using.

2 – Better account penetration and customer loyalty are critical to strong profitability and revenue growth. The right electronic licensing policies significantly enhance customer loyalty and account penetration.

3 – More efficient pricing expands your available market and allows you to develop new types of users.

4 – More effective product evaluation programs improve sales productivity and reduce the length of the sales cycle. In many markets, customers don't buy until they try a 'hands-on' evaluation.

5 – Lower manufacturing and distribution costs. The problem with software manufacturing is making CDs, shipping, handling returns and tracking SKUs. Electronic licensing decreases the number of SKUs and lowers distribution costs.

6 – Leverage one sale into the next. Electronic licensing lets you use the same CD to sell multiple products into your customer base. You can also leverage the Internet to provide more timely updates and better service to your customers.

So if you're interested in improving your software sales and profitability, talk to GLOBE*trotter*.

For a FREE evaluation kit or to learn how software publishers are using electronic software licensing successfully, visit our website at www.flexlm.com, or call us at 888-326-6476.

GLOBE*trotter*

Electronic Commerce For Software™

http://www.globetrotter.com
Email: info@globetrotter.com
Telephone: 408-445-8100
888-326-6476
Fax: 408-445-7760

FLEX*lm* and GLOBE*trotter* are registered trademarks and "Electronic Commerce for Software" is a trademark of GLOBE*trotter* Software. All other trademarks are the property of their respective owners.

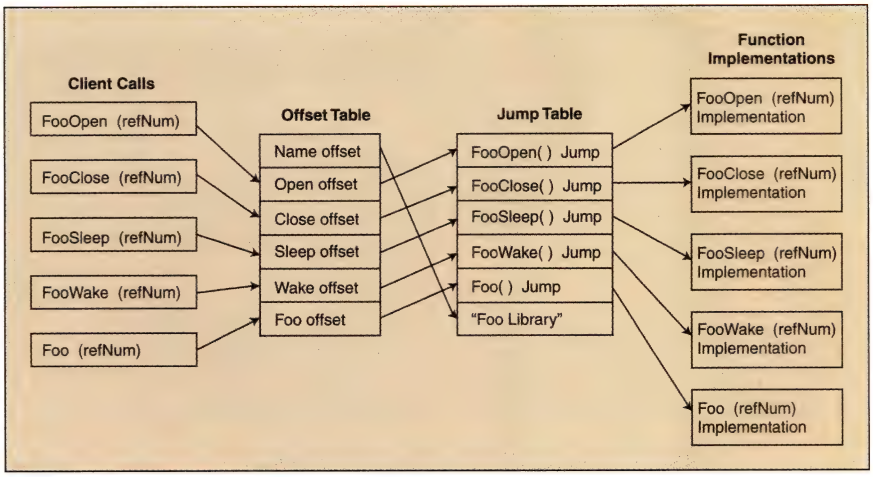


Figure 1: *Conceptual model of a dispatch table.*

The longest continuously advertised C product in the history of . . . mankind.

PC-lint for C/C++ Version 7.5 presents Bug # 520

```
#include <iostream.h>

int sum( int a[], int n )
    {
    int i;
    int s = 0;

    for( i = 0, i < n; i++; )
        s += a[i];
    return s;
    }

int a[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int main()
    {
    cout << "the sum is " << sum(a,10) << "\n";
    return 0;
    }
```

The programmer expected this program to print that the sum is 45. Instead it prints that the sum is 0. How come? Call if you need a hint or visit our web site at **www.gimpel.com**

PC-lint for C/C++ will catch this and many other bugs. It will analyze a mixed suite of C and C++ modules to uncover bugs, glitches, quirks and inconsistencies.

Version 7.0 introduced several spectacular and revolutionary innovations in the art of program validation. Taking clues from initializers, assignments, and conditionals, variable and member values are tracked, enabling reports on potential uses of null pointers and out-of-bounds subscripts. Almost 100 standard functions are rigorously checked.

Version 7.5 pushes the envelope of lint detection still further. User-defined semantic checking of functions offers a unique language to express argument constraints, inter-argument relationships, and return values. There are 75 new messages including cradle-to-grave checking of pointers and the latest admonitions and perorations of Scott Meyers.

Plus Our Traditional C/C++ Warnings: Uninitialized variables, inherited non-virtual destructors, strong type mismatches, ill-formed macros, inadvertent name-hiding, suspicious expressions, etc., etc.

Full C++ Support - PC-lint for C/C++ is based on the ARM and is tracking the latest ANSI/ISO draft. It supports both Borland and Microsoft C/C++.

PC-lint for C/C++ $239

Numerous compilers/ libraries supported. Runs on Windows 95, Windows NT, MS-DOS (built-in DOS extender), and OS/2. 30 day money back guarantee.

FlexeLint for C/C++

The same great product for other operating systems. Runs on all Unix systems, VMS, mainframes, etc. Distributed in shrouded C source form. Call for pricing.

Gimpel Software

3207 Hogarth Lane, Collegeville, PA 19426

CALL TODAY (610) 584-4261 Or FAX (610) 584-4266

www.gimpel.com

PA add 6% sales tax.

PC-lint and FlexeLint are trademarks of Gimpel Software

(continued from page 50)

parts to this. There is the definition of the trap identifiers and the associated function declarations. The installation entry point is invoked when the library is loaded. It informs the system of the library's dispatch table and initializes memory required by the library.

The installation entry point is called "__Startup__," and must be the first function in the library's link order. It returns 0 for success, or a negative error code. *SysLibLoad* returns the error code to the client application. The system passes the library's entry in the system library table as a parameter to this function; the library sets the dispatch table attribute of this structure.

The library dispatch table contains a list of all routines in the library. This is a lookup table used by the system trap dispatcher.

Implementing a Shared Library

There are three components in a shared library: the API declarations, API implementation, and dispatch table.

The shared library publishes a header file describing the API to clients. Function declarations specify how the client invokes the API, including parameter information and return types. Trap identifiers allow the system to invoke the function using the trap mechanism on which the PalmOS relies. Result codes provide an expected set of errors for which the client should check. Domain-specific structures and constants provide additional information required by the custom portion of the API.

Every shared library must publish four standard functions:

- The client must call the *Open* function first. It allows the shared library to initialize any resources it needs. No other API functions can be called prior to this function.
- The client calls the *Close* function last. It allows the shared library to release any resources it is holding. Once this function is called, the library is in an invalid state; it must be reopened before it can be used.
- The operating system calls the *Sleep* function before the device enters sleep mode. It allows system-level libraries to shut down hardware components to conserve power.
- The operating system calls the *Wake* function when the device returns from sleep mode. It allows system-level libraries to reenable any hardware components that were shut down when the device entered sleep mode.

The shared library also publishes domain-specific functions. As a rule, these functions must be invoked after *Open* and before *Close*, to make sure the library has valid

resources. The one exception is a function that retrieves the API version of the library, which may be invoked before *Open* to ensure compatibility.

If you examine the PalmOS SDK headers, you'll notice that every function declaration includes the SYS_TRAP macro:

```
Err SerOpen(UInt refNum, UInt port, ULong
      baud) SYS_TRAP(sysLibTrapOpen);
```

For the CodeWarrior compiler, this declaration expands to a Metrowerks extension called "opcode inline" syntax:

```
Err SerOpen(UInt refNum,UInt port,ULong
baud)={m68kTrapInstr+sysDispatch
                  TrapNum,trapNum}
```

The opcode inline syntax lets you specify the 680x0 opcodes for the function's implementation. When you call an opcode inline function, the compiler replaces the function calls with the specified opcodes. This feature supports calls through the 680x0's "A-Trap" mechanism; it generates the exception, which is handled by the PalmOS's system trap dispatcher.

Invoking a call to a shared library function is a two-step process. For client code, the function declaration expands into a call to the A-Trap mechanism. This mechanism uses the library's reference number to identify the appropriate entry in the system library table; the library's reference number must be the first parameter to any shared library function. Using the trap identifier as an offset into the entry's dispatch table, the appropriate function is called.

If this all seems like a lot of work, remember that the compiler takes care of it; your job is to declare the trap ID enumeration, and to make sure you use the right trap for each function declaration.

The system defines traps for the required *Open*, *Close*, *Sleep*, and *Wake* functions. You define the rest, starting with *sysLibTrapCustom*, and incrementing sequentially.

The public API also contains the errors that the functions may return. These are based on the *appErrorClass*, which is defined in SystemMgr.h. The library functions should not return results that are not either a success or one of the API-defined errors.

In addition to the functional API, there may be structures or constants defined by the library. For example, a printing library might publish a font structure and constants for bold, italic, or underline.

The functions described in the public header are implemented in a standard C file. In this module, the SYS_TRAP macro is disabled so that the function declarations evaluate to standard C declarations.

The *Open* function is responsible for allocating memory for this information and storing it in the system library table entry for the shared library. Once this is done, it performs any domain-specific initialization.

The *Close* function is responsible for releasing memory allocated for the library's globals, and removing this value from the system library table entry. Any domain-specific clean up should be done prior to releasing this memory. By convention, this function returns 0 if the library should be removed, or an error code indicating that the library is still in use by other clients.

The *Sleep* function handles notification from the system that the system is about to shut down. This notification lets system-level libraries shut down hardware components to conserve power.

The *Wake* function handles notification from the system that the system is about to wake up again. This notification lets system-level libraries reenable hardware components that were shut down when the system went to sleep.

Because these functions are invoked by system interrupts, they may only use interrupt-safe system services and must not take a long time.

Domain-specific API functions depend on the global data allocated in the *Open* function, and so should only be invoked between *Open* and *Close*. These functions follow a standard pattern. First, they retrieve the library's global data. Then, they perform the domain-specific task they provide. Finally, they release the global data.

There are exceptions to this rule. For example, many libraries provide a function to

Section	Address	Contents	Comments
Offsets	0	20	Address of name string
	2	12	Offset to Open jump
	4	14	Offset to Close jump
	6	16	Offset to Sleep jump
	8	18	Offset to Wake jump
	10	20	Offset to Foo jump
Jumps	12	JMP Open	Jump to Open
	14	JMP Close	Jump to Close
	16	JMP Sleep	Jump to Sleep
	18	JMP Wake	Jump to Wake
	20	JMP	Jump to Foo
Lib Name	22	"FooLibrary"	Library name

Table 1: *Typical dispatch table.*

retrieve the API version of the library. Because this function does not rely on any global information, it may be called before the *Open* or after the *Close* function. Indeed, it probably should be called prior to *Open* to ensure compatibility between the client and the library.

The dispatch table implementation is provided by the *Install* and *DispatchTable* functions. The system invokes *Install* as part of *SysLibLoad*; this function invokes *DispatchTable*, which returns the address of the dispatch table. This address is stored in the library's entry in the system library table. *DispatchTable* is coded in assembler and includes the declaration of the dispatch table as well as the code that returns its address.

The first section of the dispatch table is an array of offsets from the beginning of the dispatch table. The first offset points to the library name. This string is stored at the very end of the table. The remaining offsets point to entries in the next section, which is an array of jump instructions: The second offset points to a jump to the *Open* function; the third, a jump to the *Close* function; and so on. The library name is stored as a null-terminated string at the end of the jump table.

For example, say that you had a shared library named "FooLibrary," with only one function—*Foo*. The dispatch table would look like Table 1. The system executes library traps using this table. Using the trap identifier as an index into the first section of the dispatch table, it retrieves the offset to the jump instruction corresponding to the trap. The system moves to this offset in the dispatch table and executes the next instruction. This instruction is a jump to the appropriate shared library function.

As Figure 1 illustrates, this is conceptually no different from the implementation of virtual tables under C++. The client invokes a function that is mapped to an offset in a function table. Using this offset, the compiler retrieves the address of the function to execute. The main difference is that in C++, the compiler takes care of all this plumbing transparently; with shared libraries, we must do it ourselves.

This may seem more complex than it actually is. The dispatch table in the sample library I provide here (available electronically) can be copied and does work. There are a few things to remember:

- The order of the trap identifiers as declared in the public header must match the order of the respective functions in the jump table.
- There is a macro defined in the sample, *numTraps*. This must be equal to the number of public functions declared.
- There must be one entry in the dispatch table for each public function.
- The standard functions must appear first in the jump table and in the correct order: *Open*, *Close*, *Sleep*, and *Wake*.

If your library is crashing, or the wrong functions are being invoked, these are the first things to check.

And just in case shared libraries weren't daunting enough, there are some rules and caveats you should be aware of:

- Do not use static and global values. Shared libraries are stored on the device as resource databases. The memory occupied by the library is, therefore, on the storage heap. By default this memory is protected; it can only be written to with the appropriate *DataMgr* function. A downstream effect of this is that all global or static values are, effectively, read only.

  You could use static or global variables in your shared library. But if you do, you need to allocate a writeable data segment for them, and initialize this data segment by hand. When you enter a library function, you need to set the A4 register to point to this data segment. When you exit the function, you need to set it back to its original value. You also need to do this "A4 magic" before calling certain SDK

functions, such as *MemHandleNew*. In general, it's best not to use them at all.

- Install linkage. A shared library's *Install* function is actually a macro for the library entry point, __Startup__. The system expects this function to be the first entry point into the library. Therefore, this function must be first in the linkage order. We put it in its own module, and list that module first in the Segments panel of the project.
- Library resources. Palm application resources are opened with the application at all times. If you want to show a form, or raise an alert, it is a straightforward task. Resources associated with a shared library are not left opened with the library; the library must specifically load its resource database. Consequently, it must unload the database as well.
- Debugging. Bugs are a fact of life in software development. As a result, so are debuggers. Unfortunately, the CodeWarrior debugger does not currently support tracing into shared library code. This support is rumored to be coming in an upcoming release. Until this support is available, you must embed code in your library to display errors, variable values, and so on. You can use alerts, error messages, or the emulator logging facility. On some projects, we have created a debug message console on the Palm device.

A Shared Library Example

The sample shared library I present here lets the client application store and retrieve a text string. It also provides a function to retrieve the library's API version; this is useful to ensure version compatibility between application and library. The library also implements the four standard shared library functions—*Open*, *Close*, *Sleep*, and *Wake*. While the actual functionality is straightforward, this sample provides a framework onto which you can build your own shared libraries.

Summary

Shared libraries are not hard to create. Conceptually, they are function tables that follow specific rules. The bad news is that these rules are not intuitive or apparent to the first-time programmer. The good news is that once you've created your first library, you can use it for a template and stop worrying about these rules. The best news is that you can just use the sample provided with this article.

Used correctly, shared libraries will repay the investment with greater code reuse, increased application stability, and reduced resource requirements. You might think of them as peace of mind, encapsulated.

DDJ

WinDriver™

Automates hardware testing and device driver development.

Windows 95, 98, NT, 2000
Windows CE
LINUX

No DDK or Kernel knowledge needed

Cuts development time by up to 80%

Free evalution copy
www.unipress.com/wdd

USA:
UniPress Software
www.unipress.com
1-800-222-0550

Worldwide:
KRF Tech
www.krftech.com
+972-9-8870878

Do you have
THESE
In your toolbox?

- Array, LIFO, FIFO, Queue, Set and Map classes for Containers
- Array, LIFO, FIFO, Queue, Set and Map classes for Strings (ASCII/UNICODE)
- Bit Vector Class
- Encode, Decode, Compress, Decompress functions
- CRC functions
- Blazing text search engine

Introducing

KernTools™ "Accelerated Power Programming"

- ANSI C/C++ Compatible (source c/c++)
- Printed & PDF Manuals
- Competitive Upgrade & Academic Pricing
- 30-Day Money Back

Upgrades At
$149.00

Get your copy today!
www.trandyne.com
800-473-7303

Trial Versions
Additional Products
Secure Online Ordering

TranDyne

Copyright © 1999 Transitive Dynamics, Inc. All Rights Reserved

The PowerTV Operating System

A powerful OS for set-top boxes

Morgan Woodson

The PowerTV Operating System (PTVOS) from PowerTV (http://www.powertv.com/) is designed for set-top boxes and hybrid fiber/cable (HFC) networks. It consists of PowerCore, a real-time kernel and memory manager, and PowerModules, which are libraries for communicating on the network, tuning analog and digital channels, and displaying TV-centric graphics. A resident application lives in flash memory for user settings, changing channels, and displaying a program guide. Other applications can be downloaded from servers at the cable operator's head end (servers, ATM switches, and radio-frequency equipment for a cable system). Applications can be written in C and C++ to the PTVOS APIs, or in PersonalJava with Java Native Interface (JNI) methods for TV operations. An HTML engine can be called to render HTML pages.

The first commercial deployment of PTVOS is on the Explorer 2000—the first digital cable box (see Figure 1) from Scientific-Atlanta (http://www.sciatl.com/), PowerTV's parent company. Scientific-Atlanta is presently building 500,000 boxes per quarter and has installed over 100 digital head ends with over 40 launched to paying customers.

Typical Environment

The processor on the Explorer 2000 is a 54-MHz microSPARC core. There is 4 MB of flash for the OS and resident application and from 2 MB to 8 MB of DRAM for downloaded applications and data. It is a basic core with no MMU or FPU and only a small L1 cache.

The media processor has an MPEG-2 video decoder and a graphics compositor capable of 16-bit color. There are two tuners. The in-band tuner works from 57 MHz (EIA channel 2) and up, tuning 6-MHz wide bands carrying either legacy analog channels or 27 Mbits/sec. QAM64 modulated data containing MPEG-2 programs or data. The out-of-band tuner works below 57 MHz, tuning channels that are QPSK modulated, carrying 1.5 Mbits/sec. each. Out-of-band is two way and has the advantage that data can be received even when a video program is being watched.

The system includes a smartcard slot, USB port, IR receiver for hand-held remote and wireless keyboard, and 10BaseT Ethernet port—but no mass storage. The development card has extra serial ports for a monitor and debugger, extra flash for a larger debug version of the OS, and a SCSI disk interface. PTVOS will soon be available on Scientific-Atlanta's Explorer 3000 (twice as fast as the 2000) and on a box for NTL in the U.K. Most boxes planned for the near future have performance from one to four times that of the Explorer 2000.

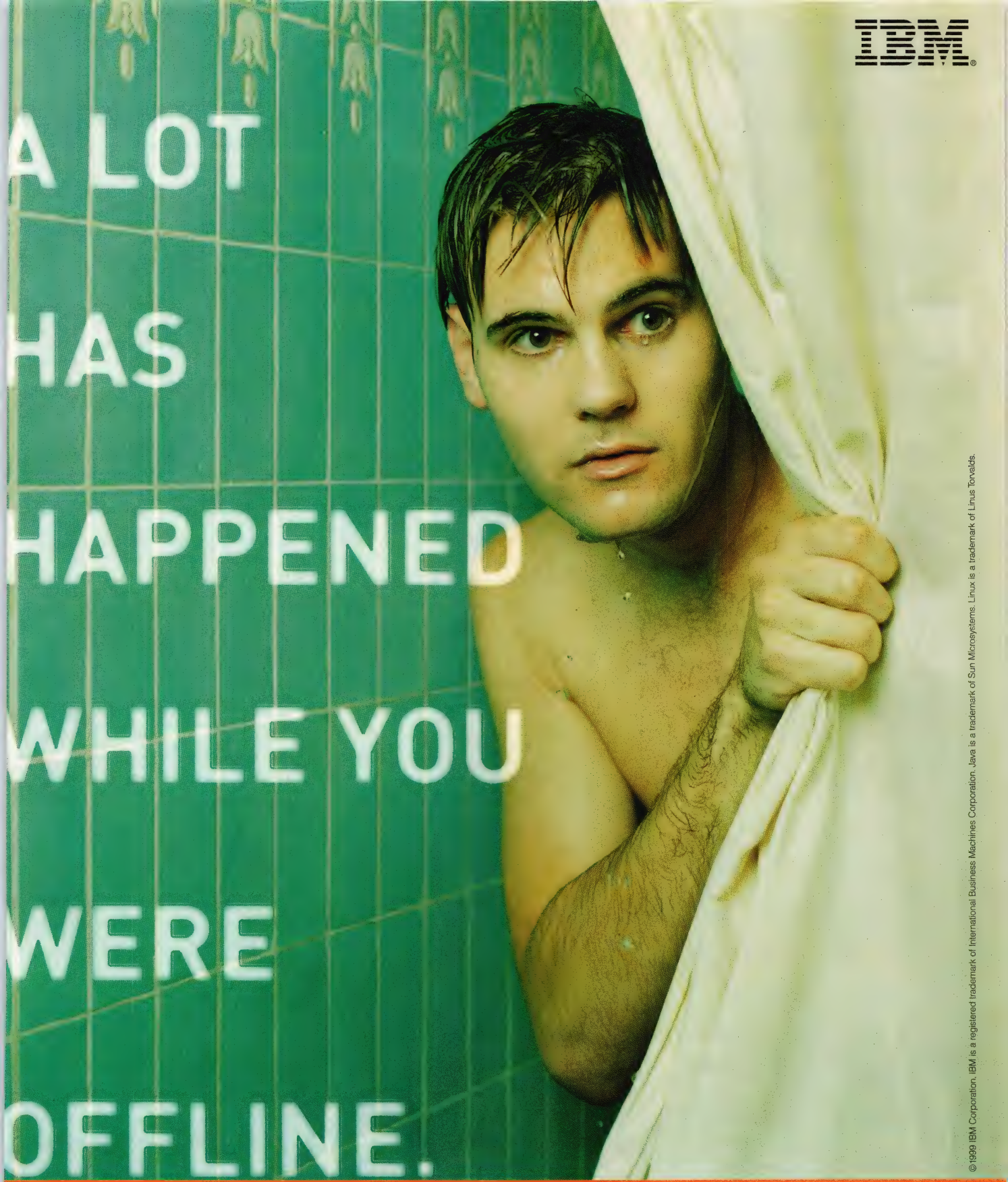
PTVOS APIs

PTVOS APIs all have a prefix (based on the name of the PowerModule) of two to four characters separated from the main function name by an underscore. For example, *pd_NewGrafPort* creates a new graphics viewport for the PowerDraw (hence *pd*) module. Structure names have the first letter capitalized; for instance, *Pd_GrafPort*; and constant names are preceded by a *k*, as in *kPd_VideoHeap*.

The PTVOS is intended for inexpensive 32-bit RISC processors that do not have MMUs so threads do not have private memory space. There are 32 priority levels; threads of the same priority run round-robin. Thread switching is preemptive, but you are encouraged to write event-driven code that calls *pk_NextEvent* early and often. Applications start with one thread each, but can spawn more as needed. For many applications, the real-time nature of the kernel will not matter. The indirect benefit is that deadline-critical tasks such as setting up new DMA transfers as old ones complete and decoding digital video can steal whatever CPU time they need without application intervention.

Communication between threads and between applications and the OS is done with *Pk_Events* in *Pk_Queues* (*pk* for PowerKernel). Events can be delivered to a single queue or posted to all queues that have registered an interest in events of that type. Events are retrieved from queues with a family of calls. *pk_NextEvent( )* returns control to the kernel. When an event is present in the calling thread's queue or the given timeout has been reached, the thread will be marked ready to run, and will run again according to its priority.

Morgan is the team lead of the multimedia group at PowerTV. He can be contacted at morganwoodson@pacbell.net.


IBM
A LOT
HAS
HAPPENED
WHILE YOU
WERE
OFFLINE.
need it? get it. tools | code | tips | how-to's | tutorials | Security | Java™ | Linux® | Web | XML
ibm.com/developerWorks
©1999 IBM Corporation. IBM is a registered trademark of International Business Machines Corporation. Java is a trademark of Sun Microsystems. Linux is a trademark of Linus Torvalds.

Need a Modern Parsing Tool??

Visual Parse++ is the only tool that brings cutting edge parsing technology to all developer's. You can design any parser without ever leaving our sophisticated visual development environment. Whether you're new to parsing technology, or an experienced veteran, Visual Parse++ is the right tool for you.

Visual Parse++ provides native programming support for **C, C++, Java, Visual Basic, and Delphi.** Our C++ and Java classes are fully supported on all platforms. Visual Parse++ also comes with drop-in parsers for **HTML, SQL, RTF, Java, C, C++,** and more.

New to version 3.0 is **natural language parsing**. Visual Parse++ is easily the most powerful parsing engine in the world with generalized parsing capability. Visual Parse++ now can handle any grammar, including ambiguous grammars. This opens up the area of **natural language parsing**, among other applications. Also included is a brand new user interface, with many new features and improvements, along with 3D parse tree and machine views.

Java | C++ | Delphi | ActiveX | VB

Order a copy today!

Call 800-988-9023

www.sand-stone.com

SandStone Technology

SOFTWARE Development PRODUCTIVITY AWARD 1995

939 Coast Blvd
Suite 4C
La Jolla, CA 92037
Phone: (619) 454-9404
Fax: (619) 454-9467
willd@sand-stone.com
www.sand-stone.com

(continued from page 56)

A complete C library is included with the OS, including *malloc* and *free*, but the memory manager provides more features if called directly. There are two basic heaps — system and video — but applications can create subheaps. They can also allocate memory with handles, which helps prevent fragmentation. Handles are an indirect reference to memory, which must be locked to get a real C pointer to the memory. The handle refers to a small block of memory that contains the real pointer and is fixed in place. While unlocked, the real data can be copied around to facilitate compaction. Compaction is automatically employed when an allocation is about to fail but can also be called explicitly.

Applications are built using GNU's make, C/C++ compiler, and linker, supplied by Cygnus Support (http://www.cygnus.com/success/powertv.html). The OS is called through a dispatch table so applications need not be recompiled for different versions of the OS. Code in the OS may move around, but the entry points in the dispatch table remain constant.

Window Manager and PowerDraw

The Window Manager provides applications with controlled access to the graphics frame buffer that sits on top of the video plane. It also filters user input events. It doesn't enforce a look-and-feel or provide window selection and rearrangement services, but does clip drawing giving the common stacked look to the windows.

Window manager also provides event routing using the window stack. Events are sent to the top window, which can act on an event as necessary. It then either passes it on to underlying windows or consumes it so the other windows never see it.

Windows that are partially or fully covered by other windows are considered obstructed. Drawing to fully obstructed windows is a no-op. Drawing to partially obstructed windows is handled by automatically repeating the drawing operation in each rectangle in a list that represents the visible portion of the window. This is a bit slower than drawing to nonobstructed windows, so you can decide only to update windows when they are on top. When a window does move to the top, it receives an expose event and must redraw all of its content. The exception is private windows, which have an offscreen buffer exactly the size of the window. Drawing goes to the buffer, which is automatically copied onscreen when the window is exposed. They are more convenient at the expense of precious memory.

Figure 1: *The Explorer 2000 unit connected to a TV.*

Another special type of window is the backdrop. This is a window the size of the entire screen, underneath all other visible windows, that determines the color format and resolution of the screen. It also determines the overlay status of the screen. Overlay is a combination of chroma keying (makes one or a range of colors completely transparent) and alpha blending (makes graphics translucent). It must be turned on once you have tuned a channel to see the video.

PowerDraw provides all the basic drawing primitives you expect: lines, rectangles, circles, ellipses, arcs, and polygons. PowerDraw is also used for rendering text through the Font Manager. New font engines can be plugged in, but the included bitmapped font engine and Agfa scalable font engine are usually sufficient.

While PowerDraw uses the concept of a drawing context, containing things like the current brush, linewidth, and antialiased drawing status (on or off), you aren't forced to use it. The context is initialized to a default state of no brush with a line width of one and once you have a *Pd_GrafPort* to draw into, you are all set. Lifting the floating cursor so it will not mangle graphics and iterating through clipping rectangle lists is handled automatically.

Usually the *grafport* is obtained with a *win_GetGrafPort* call, but you might create your own *grafports* for a couple of reasons. First is creating off-screen *grafports* as scratchpads to render bits of graphics before blitting them on screen using *pd_CopyGrafPort*. Second is when you include your own previously rendered graphics. You might have an icon of a frog on a black background that you want to hop around the screen. A 24-bit Windows bitmap of the image is converted to a 16-bit image, then to an initialized const array in a C file using PowerTV tools. You include this in your build and your code can allocate a *grafport* with no memory attached, then attach the array. Finally, blit the image on screen with the chroma key color set to black so only the foreground appears.

Broadcast Filesystem

Stream Manager provides a uniform way to read and write data from stream devices

such as FTP, HTTP, files, and serial ports. Its use is so pervasive through the system that even nonvolatile memory (NVM) and RAM or Flash can be accessed as streams. Streams to be opened are specified using the Uniform Resource Locator (URL) format. Once open, *strm_Read*, *strm_Write*, *strm_Seek*, and *strm_Close* are similar to their C standard library namesakes. Just like *_read* or *fread*, *strm_Read* takes the ID returned by *strm_Open*, a pointer to a buffer to stuff data into, and a count of bytes desired. An important difference is that you can choose to call open, read, and write synchronously, which does not return until done, or asynchronously, which returns immediately and delivers an event to the given queue when the operation completes.

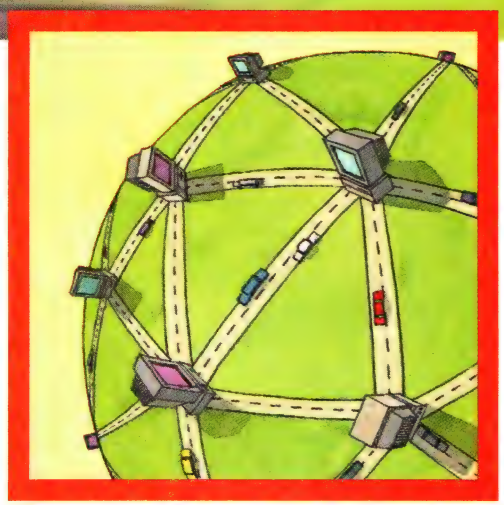
strm_Get and *strm_Set* allow stream devices to have custom parameters. The first and second parameters specify the device and parameter and the third is a *void**, allowing any kind of data to be passed or retrieved.

While PTVOS includes a full TCP/IP stack and an RF interface for communicating over the cable to the head end, two-way networking is discouraged. Two-way uses 1.5 Mbits/sec. out-of-band channels. This bandwidth is shared by all subscribers in a neighborhood. While QPSK modulators can be added to a system to divide the subscriber base into smaller groups and provide more effective bandwidth, this is expensive. One-way is preferred except for applications that absolutely require it, like signing on to the system at boot up, buying pay-per-view movies, and sending e-mail.

The downstream bandwidth is so high (27 Mbits/sec. per 6MHz-wide channel or 1 to 2 Gbits/sec. total) that many applications that require two-way on the Internet can use one-way networking in a cable system. A broadcast carousel is a stream of data on a channel that repeats as soon as it finishes. The data is divided into blocks, each with a unique address. Rather than requesting data using FTP or HTTP over TCP/IP, you request the blocks you want and the OS grabs them out of the stream when they come around. There is hardware filtering and DMA to help the OS so it does not have to consider every bit in a 27 Mbits/sec. stream.

This system of simulating choice by snatching pieces of a large database fulfills some of the ideas of push technology that was so hot a while back. Push may not be appropriate for the Internet, but works well on digital cable and on pagers. For instance, all of CNN's current global and national stories and your local newspaper's stories with pictures and weather maps can be rotated through in a few seconds. If the "Digital News" channel gets so large that response time suffers, it can be split into multiple channels.

Web-to-Host Integration



Looking for a way to integrate data from your mainframe, midrange or other host with the Web?
ViewMax is a powerful tool that allows you to preserve and extend your company's current IT investments.

Need quick access to your hosts?
ViewMax INSTANT-ON capability converts your "green screens" to interactive HTML upon installation.

Looking to build a simplified web GUI, integrate electronic commerce with your legacy systems, re-engineer your business process, or integrate multiple hosts to one interface?
ViewMax is a rapid implementation solution that allows you to bring as much or as little legacy data as you require to the web – without any modification to your host systems.

ViewMax is designed for one purpose – **TO WEB ENABLE YOUR ENTERPRISE FAST!!** But don't take our word for it. Put us to the test – we'll show you how your existing mainframe and midrange applications can be web enabled in a matter of weeks, *not months or years*. For more information call **888-846-3045** today.

Find Out How You Can Web Enable Your Enterprise Now!
www.modcomp.com/viewmax
Code: DDJ991

ViewMax is the complete solution to your host of problems.

ViewMax™
Web-to-Host Integration Tools
www.modcomp.com/viewmax

MODCOMP ©1999. All rights reserved. MODCOMP is a registered trademark and ViewMax is a trademark of MODCOMP, Inc.

The problem with a broadcast carousel is that it is a fairly low-level system. To tune a carousel, you must have control of the tuner, know what frequency the carousel is transmitting on, and know the number of the block of data you want. The broadcast filesystem (BFS) is built on top of broadcast carousel to make it look to an application more like a regular disk filesystem.

A listing of all files available through BFS is broadcast periodically with location and version information. The OS does not read the directory for the first time until a little while after it is booted. Applications can be launched before this, so applications should check whether the directory is available before opening BFS files; use *strm_Get("bfs:///", kBfs_DirectoryAvailable, (void *) available);*, where *available* is a pointer to a Boolean.

Applications cannot read the BFS directory directly, but the directory typically contains many files that are of no interest to applications such as OS update images and security and configuration information. Filenames must be hard coded into applications, but a file containing the names of files a particular application needs could be part of a carousel. For instance, a news directory file would contain the filenames of all news stories, their associated picture filenames, and English names of the stories for presentation to users.

Files on BFS have version numbers so an application can tell when they change using *strm_Get* with the value parameter set to *kBfs_FileInfo* for unopened files or *kBfs_StreamInfo* for open files. This allows the servers at the head end to passively update files without having to broadcast messages to or process status requests from set tops.

A Weather/Traffic Camera Viewer Application

Interruption of national morning news broadcasts for local news/weather typically happens every half-hour—and always when I am in the shower or in the kitchen getting breakfast. I commute on a road that snarls often and the weather in Cupertino, California, is often warmer than at my house in Santa Cruz, so I want this information. Booting a computer and dialing an ISP takes too long. What I want is to flip on the TV and get pictures from the local weather and traffic cams with no waiting.

The sample application (see Figure 2) I present here (source code is available electronically; see "Resource Center," page 5) assumes a server at the head end that can retrieve pictures from web servers, scale them for display on TV, convert them to a format PTVOS can deal with, and move them to the BFS server. The app uses PowerTV's proprietary IMG format—a 16-bit per pixel, uncompressed bitmap with a minimal header. A real-world app would use MPEG-2 I-frames because the hardware can decompress them in less than 1/60th of a second. This app scales well: A picture retrieved once from a web site can be broadcast to thousands of set tops without taxing bandwidth.

The main features of the application are as follows:

- A main loop that handles application, window, system, and user events.
- A periodic update event.
- A state machine per cam window.
- One read thread launched for each cam window.

An application has many responsibilities to be a cooperative member of the set-top society. It must acknowledge "are you there?" events that the Application Manager sends to detect hung threads, its windows must deal with changes to the window stack, and it must be ready to give up resources if they start to run low. This app acknowledges the mandatory events, but to keep it simple, does not cope with things like another application changing the color format.

After the *Init* function is called, Application Manager calls the *main* function. The app now has its own thread, but cannot draw to the screen or even create windows until it gains focus. All of the stages of coming on-line are handled by sending the *main* function events. *main* ends up being a loop that repeatedly gets an event and dispatches it. Each event causes some work to be done, but there is little simple procedural code.

The state machines keep track of whether a window is idle, reading an image from BFS, or waiting for a newer image to be placed on the server. Some of the state transitions are due to user interface events and some are caused automatically by the update event or a read finishing.

Once the app has come into focus and created and drawn all of its windows, it begins sending itself a periodic update event. This is mainly to know when to recheck the version of all BFS files that are waiting for a new one to appear. However, some state transitions are handled by switching to an intermediate state directly because of a user interface event and moving to the real state when the next update event is received. I could have created a separate thread for updates, but using a periodic event demonstrates how to weave two threads into one using events.

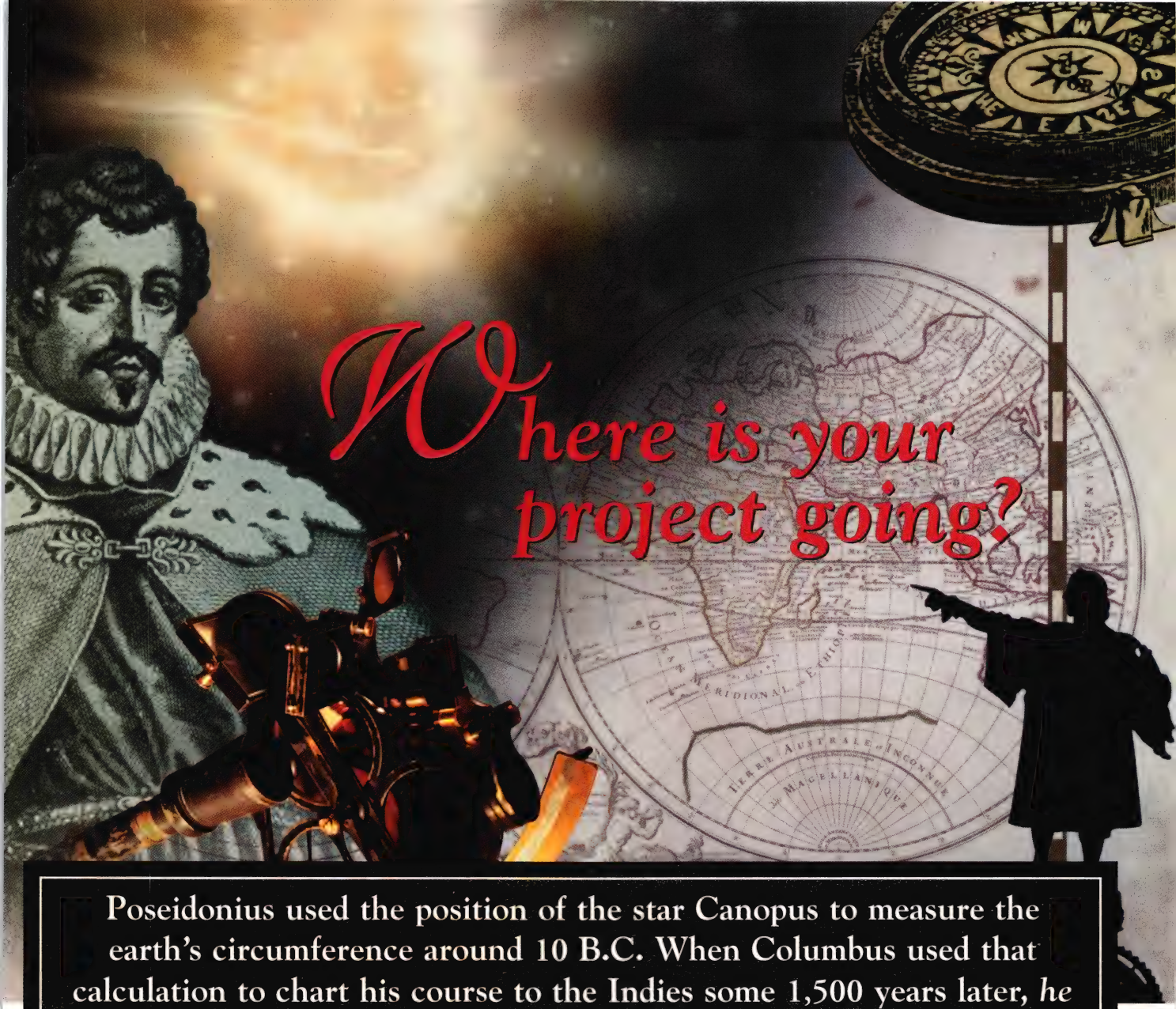
Stream manager functions can be called asynchronously so they return immediately and deliver events when the operation completes or fails, but the update event had already made things complicated and I wanted to demonstrate how easy it is to launch additional threads from an app. A function launched as a new thread must have a particular prototype, but beyond that it can do anything it wants. It can be pending on events from a queue or just run as a loop. When the work is done, the thread function simply returns and the thread context is destroyed.

Conclusion

If you are familiar with UNIX/X, Windows, or Macintosh, the PTVOS should seem familiar with its standard C library, TCP/IP networking, and a simple drawing library. However, things start getting interesting when you write television-centric applications with overlay over video, cable-specific networking, and a nonkeyboard user interface.

DDJ

Figure 2: *Running the weather/traffic view application.*



Poseidonius used the position of the star Canopus to measure the earth's circumference around 10 B.C. When Columbus used that calculation to chart his course to the Indies some 1,500 years later, *he ended up off-target, over-budget and late.* Though adequate for shorter journeys, it wasn't the best tool for a complicated, long-range voyage.

Still using a relational database with OO programming? Where is your project going?

Using a relational database management system with object-orientated (OO) programming can severely undermine the benefits of OO. The costs of mixing object-to-relational programming are high and often negate the benefits of OO programming, such as flexibility, reuse and simplicity. Using an object database with OO programming can reduce your code by 30%.

To learn more about how Objectivity's object database can help you exploit the object-related benefits in your application, contact us for a free copy of our white paper, ***Accelerating Your Object-Oriented Development.***

To get your copy, and discover a whole new world, visit our web site: **www.objectivity.com** or call (800) 767-6259.

Objectivity

© 1999 Objectivity Inc. Objectivity and Objectivity/DB are registered trademarks of Objectivity, Inc.

The BeRays Ray Tracer

Making it easy to plug in new rendering ideas

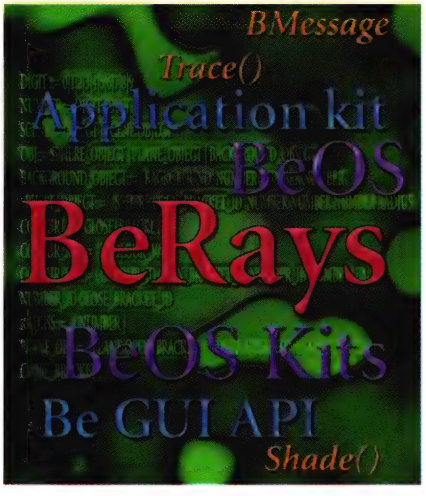
Regan Russell

BeOS, an operating system from Be Inc. (http://www.be.com/), was designed for digital media applications on PCs and Internet appliances. Consequently, BeOS supports a 64-bit journaling filesystem, real-time multimedia streaming, pervasive multithreading, and multiprocessor support.

There are two types of Be applications—GUI applications and console applications. A console application can be written in C and is comparable to UNIX programming. In fact, many tools are GNU based. The default shell is the BeShell, a slightly modified Bourne Again Shell (BASH), popular with users of FreeBSD, Linux, Solaris, and the like. The open-source versions of Lex and YACC—Flex and Bison—are also available. Additionally, for the Intel version, NASM (assembler with MMX support) is available.

The Be GUI API, on the other hand, is C++ based. Unlike Windows (and XII), which lets you write apps in C using the raw Win32 API without using C++ and MFC (or Motif applications without a C++ wrapper), you cannot write GUI-based BeOS applications in C. To do anything interesting, applications must create objects derived from the classes *BApplication*, *BWindow*, and *BView* in C++.

In this article, I present a GUI-based Be ray-tracer application called "BeRays"—an object-oriented application (written in C++) that makes it easy to plug in new rendering ideas. Most public source ray tracers are written in C and simulate inheritance in case statements or by implementing their own virtual function tables (commonly known as "vtbls") by manipulating function pointers. In BeRays, I use flex and bison to generate the parser for the input language.The source code and related files for the BeOS implementation of BeRays is available electronically; see "Resource Center," page 5. Although I've tested BeRays on both Intel BeOS 4.5 and PowerPC BeOS 4.0, I'm only providing the Intel version with Intel project file. To build a PowerPC version, you simply create a new project and add all the required source files and libraries.



BeOS Kits

The keys to developing BeOS applications are kits (classes and interfaces to classes that provide support) and the Smalltalk Model View Controller (MVC) concept. There are a number of kits, including:

- The Application kit. The most important kit for beginning BeOS programming is the Application kit. Like most C++ programs, a BeOS application starts in *main()*. The *main* function is usually used to create an application object and *Run()* it; see Listing One. When using a derived class version of Listing One, the subclassed application object constructor usually constructs a window and view, then attaches the view to the window and the window to the application. The MIME encoded string is a signature used by the constructor (*BApplication::BApplication()*). There is one string of this type for each application. The operating-system framework uses the signature string in the same manner that MIME is used in browsers. It knows to run applications. Windows COM experts might consider a Be application's signature similar to a GUID in that it distinguishes components, or in this case applications, from one another.

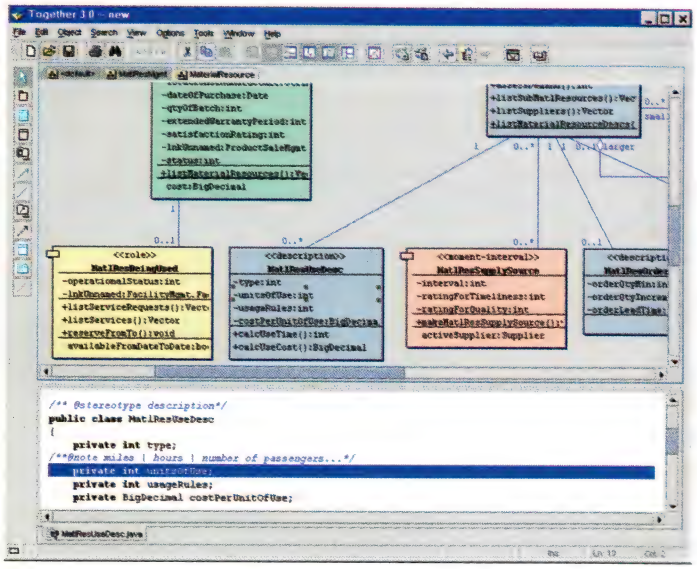
  Additionally, the Application kit includes *BMessage*, which provides functionality for application objects and is the basis for event handling. Any class derived from *BHandler* or *Blooper* can handle *BMessages*. In addition to normal events, such as a window being resized, messages can be sent from one application to another. This is called "scripting" and supports chaining graphical applications together in the same manner that shell commands are linked.
- The Storage kit handles file I/O, directories, MIME types, and filesystems.
- The Interface kit handles all GUI work and contains drawing, controls, view, and window classes.
- The Kernel kit facilitates low-level access to kernel features such as threads, teams (a collaborative collection of threads), semaphores, and linker images.
- The Media kit provides an interface to sound files and digital-to-analog (and back) conversion support.
- The Midi kit supports Midi sampling and synthesizer files.
- The OpenGL kit provides an interface to OpenGL support of GLScreens and GLViews.
- The Device kit handles joystick, the three-fuse geek port, and serial I/O.
- The Game kit enables direct window and direct screen access for games.

Regan is a programmer in Sydney, Australia. He can be contacted at rrussell@c2.telstra-mm.net.au.

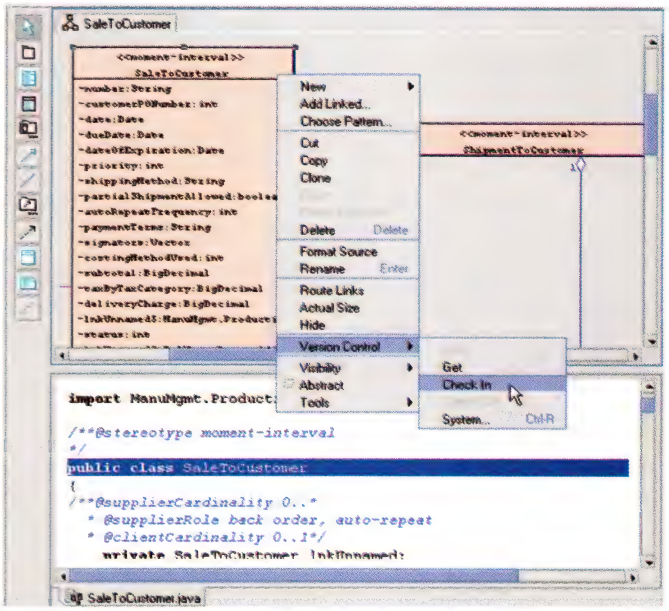
New!

Together/J's EJBs, GoF patterns, and simultaneous round-trip engineering—what Home Depot calls "the backbone of all our Java development"

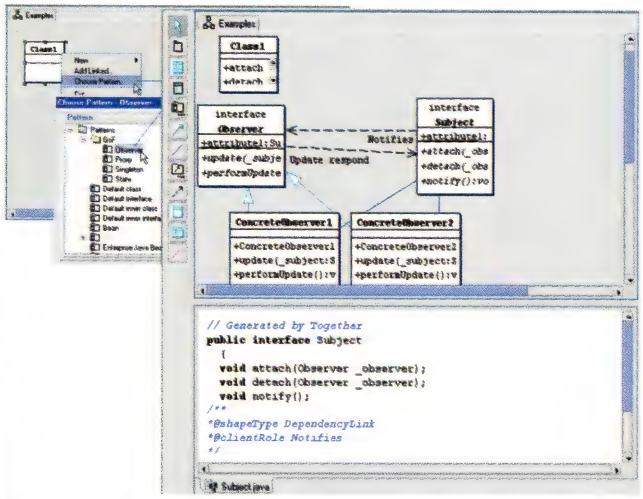
Download the free TJ Whiteboard and experience TJ for yourself



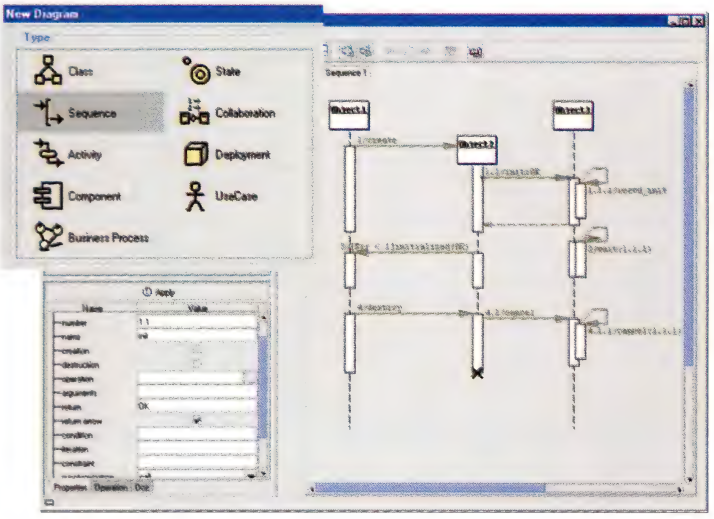
Simultaneous round-trip—always in-sync always up-to-date, complete freedom to work visually or textually as you see fit



Version-control features are a just mouse-click away—complete end-to-end versioning of both models and source across your entire team



Select a class, choose EJB™ or a GoF pattern, and TJ auto-generates iit for you – in Java or C++ (or both, in Together/Enterprise)



UML 1.3 (nine kinds of diagrams) including reverse engineering of methods into sequence diagrams. Plus HTML and RTF documentation—invoke it as part of your daily build, deliver always-up-to-date doc to your boss

TogetherJ

Download TJ Whiteboard from www.togethersoft.com/dd/

"Having seen Together/J's direct use of Java code to support the models, I am forced to wonder why anyone would like to work in any other way! I do think that the strength of the product lies in its interoperability — it's a 'complete solution', not just a modeling tool." — Jon Collins, Senior Analyst, **Bloor Research**

"I really like Together/J. No other vender has even come close to making such an intuitive and convenient diagramming interface." —David Marine, Sr. Web Applications Developer, **NetSolve**

"We discovered in our evaluation that other modeling tools only get used during the front-end of a project. Yet we found Together/J to be the one UML modeler that our developers will use from start to finish. Why? It's the only one that delivers simultaneous (rather than batch-mode) round-trip engineering." — Curtis Chambers, **The Home Depot**

"I'm sold... I just finished doing the entire Use Case diagram for the project, (in the format I wanted it), publishing it to a web document on our web site, am working on the sequence and collaboration diagrams right now; and am scheduling a net video discussion group of all our users to discuss the use case designs that we posted... All with Together/J. Your product does everything we need to do now and more..." — Steve McConnell, **The Shaw Group**

"Together/J is an excellent modeling CASE tool for Java developers... It also meets the needs of real-world object modelers. Its support for language-specific features such as inner classes and JavaBeans, puts it miles ahead of other CASE tools that merely support the Java-language syntax... Together/J 3's support for key UML 1.3 diagrams, patterns, and Java-specific language features makes it a valuable addition to your Java development toolset." — **Jolt Award '99**

TogetherSoft L.L.C.
1720 Leigh Drive
Raleigh, NC 27603 USA
(919) 772-9350
www.togethersoft.com

rated jars top 1%

Editors' Choice JavaWorld 1998

Software Development Product Excellence Award

©1999 TogetherSoft L.L.C. All rights reserved.
Together is a registered trademark of TogetherSoft L.L.C.
Java and EJB are trademarks of Sun Microsystems, Inc.

(continued from page 62)

- The Network kit supports sockets and network services.
- The Translation kit enables the development of filters for different file types; for example, PNG and BMP.
- The Mail kit is a foundation of e-mail agent and delivery services.
- The Support kit contains miscellaneous functionality including void pointer lists (*BList*) and profiler watchdog support.

Kits talk to servers. Servers are daemon processes that manage things. One such server is the application server, which includes a psycho-killer thread that goes around cleaning up processes that have gone wild, such as UNIX zombie processes. Kits also require a library and possibly a header file to be included. For example, the OpenGL kit needs libGL.so and be/opengl/GLView.h.

Kits are not only threadsafe but spawn threads themselves. Creating a window (from the Interface kit and managed by the application server) creates a separate thread to manage it. A group of threads are a team and an application is usually a team of a few threads.

Many objects in kits take responsibility for their subobjects. For example, a *BMenu* will delete each *BMenuItem* in its destructor. It is often valid to allocate variables without keeping track of them and not expect memory leaks. BeRays contains a few new statements without matching deletes. The general case still remains that if the OS allocates it, the OS frees it. If the application allocates it, then the application should free it if the OS will not. *BList*, for example, will not delete attached objects, but will delete its list nodes.

Messages are generated from events. Events can have their handler specified at creation time. For example, the *BFilePanel* in BeRay's ray tracer directs its messages to the parent window. The messages the parent receives depend on the user choices when the dialog is shown. Some events are handled by member functions for example *BWindow::FrameMoved()*. Threads do not receive messages. Message handlers receive messages.

The BeRays Ray-Tracer Application

The major components of the BeRays ray tracer I present here are the user interface, rendering engine, and rendering-engine support. The *main* function fires off the user interface components, and the rendering engine is then constructed and rendered (via the Go menu item, handled by the *MessageReceived* member function).

The application starts in *main* and creates a new RaysApp application, *Runs* the application, and then deletes the application when the *Run* returns. The RaysApp's (subclassed from *BeApplication*) constructor creates a *RaysWnd* (subclassed from *BWindow*). The window handles messages and the constructor creates a view. The RaysView constructor creates *BMenuItems* and a *BMenu* and attaches the menu items to the menu and the menu to the view. The message is directed to the window message handler when the menu item is constructed.

The *Go* member is called from the window's message-handling function. The *RaysWnd MessageReceived* member is called when there is a MOPTION_RENDER message generated from the menu item. The menu item is created in the window's constructor with *BMenuItem('Render', new BMessage(MOPTION_RENDER));*.

Another message the windows message handler deals with is the file open menu message. The file is read and input is passed to the flex scanner and bison grammar parser. The parser calls the *MultiRender* member to create new objects. Listing Two is the BNF grammar for the input language.

The general approach I take to ray tracing is not new and is well documented by Glasner, Foley and van Dam, Heckbert, Watt, Watt and Watt, and others, including *DDJ* (see, for instance, "RAY: A Ray-Tracing Program in C++," by Alain Mangen, July 1994; and "Ray Tracing," Daniel Lyke, September 1990). It is fairly well understood that backward ray tracing starts at an eye or camera position and rays are traced through a view plane toward a scene full of objects. If the ray intersects an object, other rays are traced from the surface of that object for shadows, shading, and reflection. If the ray does not intersect with an object, a background color is set. Forward ray tracing starts at a light source. Backward tracing is also known as "persistence of vision."

Because a photon is its own antiparticle, a ray going forward in time from a light source to a light receptor can be considered the same as its antiparticle (itself) going backwards in time from the receiver to the transmitter. This has the added bonus that rays emitted from the source but never reach the receiver are not considered. This is a significant speed improvement. For a photon travelling at the speed of light, time has no meaning.

For all rows and columns of the image, the *Go* member function constructs a ray starting from the viewpoint and starts a (potentially recursive) *Trace* through the scene, determining intersections of the ray with scene objects. The result of the *Trace* is passed to the *Shade* routine, which determines a color (possibly by calling *Trace*) and passes it back up to *Go* where it is set. The *RenderEngine* also maintains a list of scene objects and a list of light sources.

The *Trace* iterates through a list (*BList*) of scene objects and determines the rays intersection (or lack of) by using the scene objects intersection function. The *Shade* member uses the scene objects *CalcNormal* interface to determine the amount of shading. In fact the *Trace* and *Shade* members directly or indirectly use a lot of support functions.

SceneObject is a base class for any objects that appear in a scene. *SceneObject* is used in *Trace*, *Shade*, and LightSource's *GetColour* member for shadow determination. Scene objects need to support three interfaces: Construction, Intersection, and Calculate Normal. Light source, sphere, and plane are types of scene objects. New objects and variations of existing objects can be added at will. Scene objects store properties as a surface object.

Light source is a type of scene object and is used in the shade routine. It contains the light's color and position and has interfaces for getting the color and ray intersection. The method for getting the color determines if there are objects between the light source and the object the traced ray has stuck. This is used in shadow determination and is called a "shadow feeler."

In addition to scene objects, the *MultiRender* class uses a color class to represent colors such as the light to define a scene. The color class stores red, green, and blue triples. It is used in shading and specifying objects properties. Colors can be specified in the range of 0 to 1.0 or in 0 to 255. They are automatically converted.

Vectors are used extensively through the render engine to generate an image from the definition. The vector class is a mathematical entity for storing x, y, and z triples with operator overloading for vector operations. The methods and operations include length, rotate, reverse rotate, min, max, ~ (normalization), ^ (cross product), % (dot product), * / +, and –. A ray is two vectors from an origin to a direction. All vectors only contain a direction as they start from a common origin of (0,0,0).

The generation of the image starts with the viewpoint. The viewpoint is an eye position or camera position. A viewpoint contains four vectors: up, right, location, and direction. Rays originate from the viewpoint and once constructed with MakeRay, the viewpoint is discarded.

With future development in mind, I defined an image class with a red, green, and blue channel bitmap so that per channel image processing could be done later.

To build BeRays, you start at the command line and compile the lex (flex) and yacc (bison) specifications into C sources with the *make* command and a standard Makefile. The project can then

be compiled and linked with the Build menu item from the IDE.

When running BeRays, you are presented with a window, blank white view, and menu bar. The menus include File, Go, and Help. The file menu consists of Open scene file and Quit. The Go menu contains the item Render and the Help menu contains About. The scene file is created with a text editor like vi. The program on start up does not contain a scene, so a default scene is used.

Conclusion

BeRays is simple and can be extended in a multitude of ways. Different illumination methods could be tried, as well as adding bounding volumes, BSP trees, and octrees. One major problem to solve is spatial aliasing or jaggies. Antialiasing techniques (such as stochastic ray tracing) could also be applied. Image processing could be integrated. Other types of objects can be added: You need to create a derived class for the new object and override the *Intersection* and *CalcNormal* members and add lex and yacc descriptions for the new object and a function in the render engine to add the object passed in from the parser to the scene list.

References

Glasner, Andrew S. (editor). *An Introduction to Ray Tracing*, Academic Press, 1989.

The Be Development Team. *Be Developer Guide*, O'Reilly & Associates, 1997.

Heckbert Paul S. (editor). "A Minimal Ray Tracer" Graphics Gems IV, Academic Press, 1994.

Foley, James D. Andries van Dam et al. *Computer Graphics: Principle and Practice*, Second Edition, Addison-Wesley, 1990.

Watt, A. *3D Computer Graphics*, Second Edition, Addison-Wesley, 1993.

Watt, A and M.Watt. *Advanced Animation and Rendering Techniques: Theory and Practice*, Addison-Wesley, 1992.

DDJ

Listing One

```
#include <Application.h>
main()
{
   const int left = 10, right = 500, top = 10, bottom = 200;
   // Create application, window and view..
   BApplication *TheApp = new BApplication( "application/x-vnd.ddj-sample" );
   BWindow *TheWindow = new BWindow(BRect( left, top, right, bottom ),
            "Rays", B_TITLED_WINDOW, B_NOT_ZOOMABLE | B_NOT_RESIZABLE );
   BView *TheView = new BView( BRect( 100, 100, 200, 200 ),
            "Rays", B_FOLLOW_ALL_SIDES,  B_WILL_DRAW );
   // Attach view to window, show the window and run the application.
   TheWindow->AddChild( TheView );
   TheWindow->Show();
   TheApp->Run();
   // Clean up.
   delete TheView;
   delete TheWindow;
   delete TheApp;
}
```

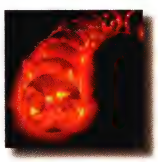
Listing Two

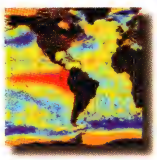
```
DIGIT ::= 0|1|2|3|4|5|6|7|8|9
NUMBER::= DIGIT {DIGIT}
SCENE::= OBJECT | SCENE OBJECT
OBJ::= SPHERE_OBJECT | PLANE_OBJECT | BACKGROUND_OBJECT
BACKGROUND_OBJECT::=   BACKGROUND  NUMBER NUMBER NUMBER
SPHERE_OBJECT::=    SPHERE OPEN_BRACKET_ID  NUMBER NUMBER NUMBER RADIUS
COLOUR_REF CLOSE_BRACKET
COLOUR_REF::=   { COLOUR_VAL }
COLOUR_VAL::=   COLOUR_ID OPEN_BRACKET_ID NUMBER_ID NUMBER_ID
NUMBER_ID CLOSE_BRACKET_ID
RADIUS::=   { NUMBER }
PLANE_OBJ::=    PLANE OPEN_BRACKET NUMBER NUMBER NUMBER NUMBER
CLOSE_BRACKET
```

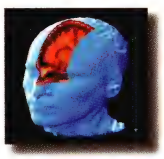
DDJ

Go Beyond

IDL
the Interactive Data Language







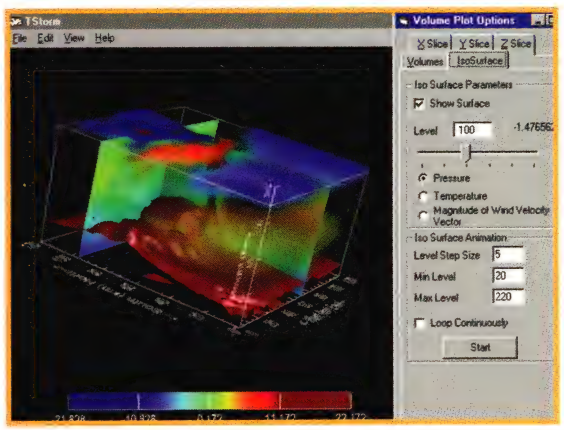
Solve Big Visualization Problems

For developers that need advanced visualizations, high-performance data analysis and development tools, IDL is your solution. IDL®, *the Interactive Data Language*, is a powerful 4GL that delivers state of the art graphics, produces cross-platform applications and handles common, emerging and user-defined data formats.

IDL 5.2 Development Feature Highlights

- ✔ IDL Development Environment
- ✔ IDL GUIBuilder
- ✔ ActiveX Control
- ✔ OpenGL Accelerated 3D Graphics
- ✔ Extensive Suite of Math & Statistics Routines
- ✔ Object-Oriented Graphics Technology
- ✔ Pre-built Visualization Applets
- ✔ Full Set of GUI Components
- ✔ Flexible Data I/O

Combining robust code editing, profiling and debugging with an extensive suite of processing routines and leading visualizations, IDL is equal to any requirements. Once you see how easy IDL is for solving big visualization problems – you'll wonder why you waited so long.



www.rsinc.com/go

Visit our Web site Today – Discover IDL

RESEARCH SYSTEMS
Software = Vision.
303.786.9900
info@rsinc.com

International Offices
Austria, Germany, Liechtenstein, Luxembourg, Switzerland, The Netherlands, Brazil, China, France, Belgium, India, Italy, Japan, Korea, Spain, Portugal, Taiwan, United Kingdom

IDL is a registered trademark of Research Systems, Inc. All rights reserved.

Designing the legOS Multitasking Operating System

An OS for the Hitachi H8 microcontroller

Markus L. Noga

Mindstorms is a system developed by Lego and the Massachusetts Institute of Technology (MIT) that lets you design, build, and program robots using standard Lego pieces, a handful of sensors, and a microcontroller that is programmable via an infrared link with a PC. More specifically, the Mindstorms Robotics Invention System (see http://www.legomindstorms.com/) consists of more than 700 Lego pieces, two motors, gears, two touch sensors, a light sensor, the microcontroller, an infrared transmitter, and a PC cable to connect it to one of your COM ports. The packaged software currently supports Windows 95/98/NT.

Constructing Mindstorms robots involves combining normal Lego pieces with sensors (see Figure 1) that can handle data input and enable the robot to respond to external situations. You program the Robotic Command Explorer (RCX) computer (Figure 2) in a graphical language called "RCX Code." Finally, as Figure 3 illustrates, the code is then downloaded from your PC to the RCX via an infrared link.

Although you typically program the RCX using RCX Code, you can also use SPIRIT.OCX (provided in the Mindstorms SDK) to control the unit using Visual Basic, Visual C++, Delphi, or the like. Figure 4 illustrates the RCX software system, which consists of a graphical programming environment, OCX control,

and bytecode Virtual Machine (VM). Although powerful, each of these layers introduces certain limitations. The graphical environment limits variable usage to one counter, rendering it unusable for any but the simplest applications. The event-driven programming model is severely hampered by the lack of reentrancy and task synchronization. And finally, like the OCX control, the environment is Windows only.

The VM, on the other hand, provides 32 global variables, a write-only datalog, eight concurrent tasks, and direct control from a PC. Still, the VM also suffers from drawbacks, lacking, for instance, a stack, a real heap, and the concept of priorities.

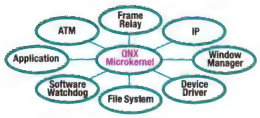
Alternative Software

Because of these and other limitations in Mindstorms software, a number of freeware/shareware alternatives have become available, since Kekoa Proudfoot reverse engineered the bytecode instruction set (http://graphics.stanford.edu/~kekoa/). Dave Baum conceived of Not-Quite-C (NQC), a C dialect suitable for the bytecode VMs. This implies no local variables, compound expressions, reentrancy, arrays, or user-defined data types. Baum's NQCC compiler, now integrated in his RCX Command Center, runs on a wide variety of host platforms and is still the most widely used bytecode cross-compiler. Other tools include RCX Creator (Barry Ruffner), BotCode (Marty Balash), and Bot-Kit (based on Dolphin Smalltalk), along with tools and techniques for programming the RCX with VBScript or JavaScript.

Programming environments centering around NQCC and other compilers targeting the RCX VM followed, and high-level wrappers for the OCX control provided by Lego abound. However, all these approaches share the limitations imposed by the VM. Frustrated with these shortcomings and the unavailability of so many crucial features, I launched the legOS project as an alternative to the RCX firmware.

Markus is a student of computer science at the University of Karlsruhe, Germany. He can be contacted at noga@inrialpes.fr.

The RTOS that answers the call:



- MMU protection for *all* processes
- Production-ready application suites
- Embedded windowing system
- Java platforms
- POSIX API
- Support for x86, PowerPC, MIPS
- BSPs

It's a fact: everyone wants more features. Even the humble pay phone now has to be a web-browsing, fax-sending, video-conferencing, life-enhancing, all-in-one-answer-to-every-problem-you-could-ever-imagine.

Just one thing. How can you possibly create embedded products that will support new features — on demand and on time — and still achieve "five nines" availability?

Call about our Eval System!
800 676-0566 ext. 1387

Start with a proven OS: QNX®. With QNX's advanced UPM™ architecture, you can dramatically **reduce the time** needed to prototype, debug, and test new software features — while making your system virtually **immune to software faults**. Better yet, QNX lets you add or upgrade features ***on the fly***. So you can deliver what customers want, when they want. Without missing a single call.

www.qnx.com

▸ **Visit us at ESC West, booth 1618**

QNX®
The Leading Realtime OS for PCs
now multiplatform

©1999, QNX Software Systems Ltd. QNX is a registered trademark, and UPM is a trademark, of QNX Software Systems Ltd. All other trademarks and registered trademarks belong to their respective owners.

(continued from page 66)

My goal was to design a small embedded operating system, implemented in standardized languages and native assembler instead of exotic dialects and bytecodes. In other words, I wanted to tear down the padded VM environment while providing similar or superior system services. legOS offers a number of features, including preemptive multitasking, energy saving, dynamic memory management, POSIX semaphores, native access to display, buttons, infrared (IR) communication, motors and sensors, and a new kernel and library. legOS can be programmed with standard C or assembler. It is available from *DDJ* (see "Resource Center," page 5) and at http://www.multimania .com/legos/.

The Real Machine

The heart of the Mindstorms RCX is a 16-MHz Hitachi H8/3292 series microcontroller. This processor line sports an 8-channel 10-bit A/D converter, serial communications interface, several timers, and multiple digital I/O ports among its on-chip modules. The version inside the Mindstorms offers 16 KB of ROM and 32 KB of RAM. Fortunately, the H8 instruction set is a supported target platform for several flavors of the freely available GNU C compiler. I used binutils-2.9.1 and gcc-2.8.1 under RedHat Linux 5.1. Pre-1.1.1. egcs and other gcc versions might need minor patches to cross-compile correctly, so stick with gcc-2.8.1 if possible.

These tools are available from any number of sources, including the GNU ftp site mirrors and most Linux distributions. Their installation on a UNIX-like platform is a near-classic case of "make ; make install;" see Example 1. Installation under Windows is equally straightforward with the aid of cygwin32, a free UNIX compatibility package by Cygnus Support (http://www.cygnus.org/).

Design Decisions

I chose to comply with standard UNIX interfaces wherever possible. Most C programmers are familiar with *puts()*, *sleep()*, *malloc()*, and *exec()*, and I wanted to ease their transition. However, this is also a question of personal taste, because I dislike the abbreviated, technical naming conventions in other embedded operating systems that might have served as role models.

Outside the VM sandbox, there is no memory protection whatsoever; like many embedded systems, the H8 simply lacks the concept. Malformed or malicious user applications can always overwrite kernel

Figure 1: *Lego touch sensor.*

Figure 2: *The RCX computer.*

Secret to device driver tools found - Support

What type of support do you want?

☑ Support from World Class Device Driver Engineers.

There's only one place to head...

➡ **www.bluewatersystems.com**

☐ Support from someone just entering boot camp.

www.commonware.big

800.962.2114

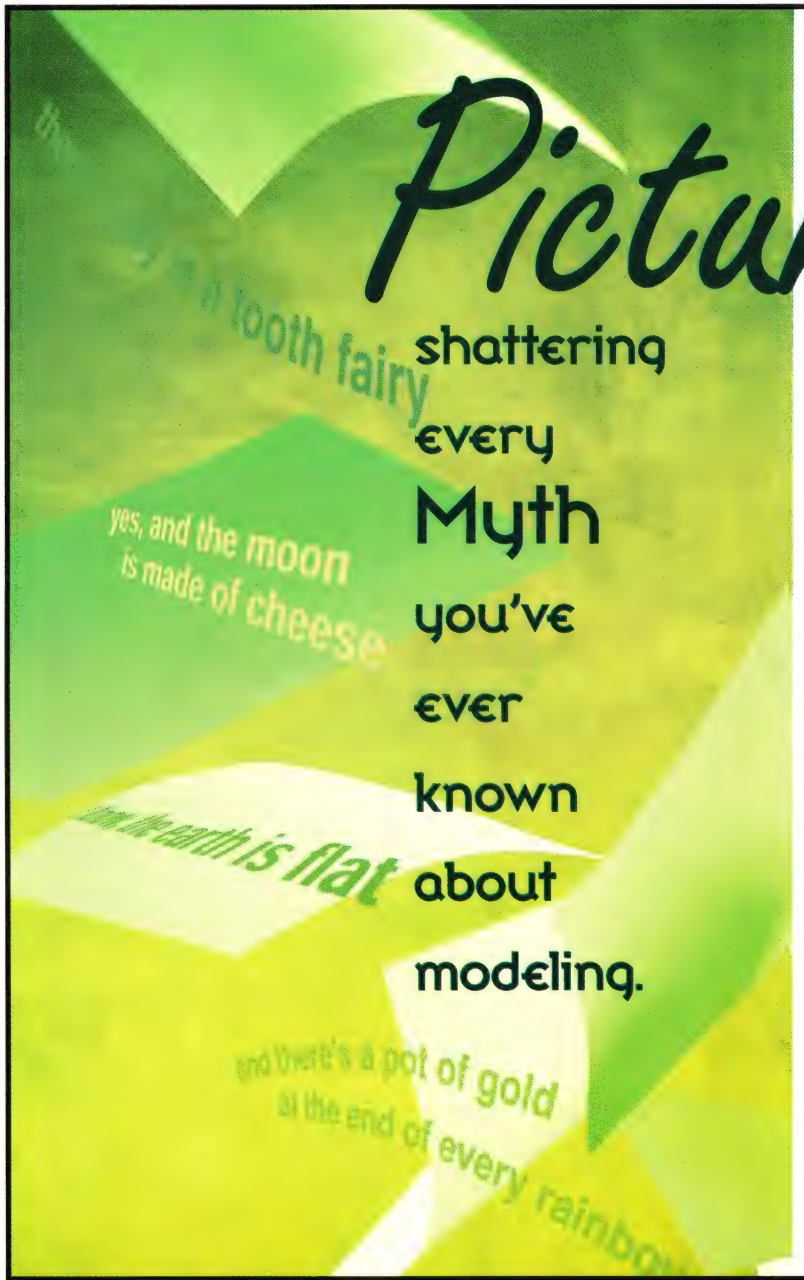
Dr. Dobb's Essential Books on Numerics and Numerical Programming CD-ROM

This comprehensive reference tool will give you:

- Detailed introduction to numerics, numerical analysis, and their applications
- Solutions to numerical problems based on limited memory storage space for data and applications
- Thorough up-to-date treatment of the behavior of numerical algorithms, precision arithmetic, combining algorithmic derivations, and rounding analysis
- Multigrid methods for solving partial differential equations
- Spectral analysis on unevenly sampled data
- Calculation of numerical derivatives
- Write efficient, high-speed arithmetic routines, and more!

Dr. Dobb's CD-ROM LIBRARY

Price: $99.95
CALL: 800-992-0549
or visit our website:
www.ddj.com/cdrom/



modeling a real-time system with tools that provide immediate benefits like system architecture, task modeling, or documentation generation.

more functionality such as round-trip engineering, timing notes, multi-user repository, real-time perspective mentor, automatic document generation.

more support for state modeling including simulation, animation, and code generation.

a tool so easy to adopt that software engineers can begin developing the first day.

a development team that remains in sync throughout the development process.

on-time integration with ease.

using the most flexible modeling tools on the market.

coming in under budget.

ARTiSAN Real-time Studio Professional and Real-time Modeler

shattering myths for as little as **$995.***

ARTiSAN SOFTWARE

*Limited time North American promotional pricing. Similar savings worldwide.

Toll free: 1-888-511-7975 • Phone: 1-503-245-6200 • +44-1242-229300 (Europe) • www.artisansw.com

(continued from page 68)
code and data. Consequently, I sacrificed security for speed and simplicity.

To minimize memory footprint, I wanted to give programmers full and fine-grained control over which functionality to include in the kernel. As in most embedded operating systems, this can easily be achieved with conditional defines.The full system's memory requirements are currently a third of the original firmware's.

The kernel was to be statically linked to user applications, forming a binary downloadable with standard firmware tools. This was the only initially feasible method, as an IR driver didn't exist. Also, it emphasizes independent operation of the RCX. Dynamic linking of user applications against a kernel symbol table on a host PC will be a future option for task download via IR.

I decided against a filesystem, since there's no mass storage on the RCX. And although user-mode hardware access via special device files is an elegant method on systems with memory protection, the operating security gained on the H8 would be illusory. Currently, I like to view the hardware devices as OS-provided objects with a thin public interface mostly consisting of inline functions. Device files might be considered in the list of possible future extensions for the sake of elegance. Then again, when did *ioctl()*s become elegant?

By textbook definition, the three basic tasks of an operating systems are:

- Memory management.
- Resource management.
- Task management.

Lacking a complete emulator for the RCX hardware, resource access was clearly the first milestone to tackle. Without I/O, it's difficult to see if your programs work. As tasks need stacks, memory management became the second milestone and task management the third. Thus, the course of development was set.

Resource Management

Due to space constraints, I'll only address the console driver—that is, the LCD display and button control—in this article. Although most of the other drivers are fairly straightforward derivations of the H8 hardware manual (see http://www.multimania.com/legos/), nearly every single one contains RCX-specific twists. For example, the IR driver needs to enable carrier frequency generation exclusively when sending, and the A/D driver has to provide pulsed alimentation for active sensors such as light and rotation.

The starting point for the console driver involved the ROM functions to display symbols and numbers on the LCD and refresh the display and query buttons. Their addresses and some of their parameters had been located by Kekoa, who used these functions in his first assembler firmware.

In an early version of legOS, I relied exclusively upon ROM to address the LCD. Examination of the ROM disassembly narrowed the region of possible control codes. I then wrote experimental programs that stepped through these ranges and noted the display changes affected.

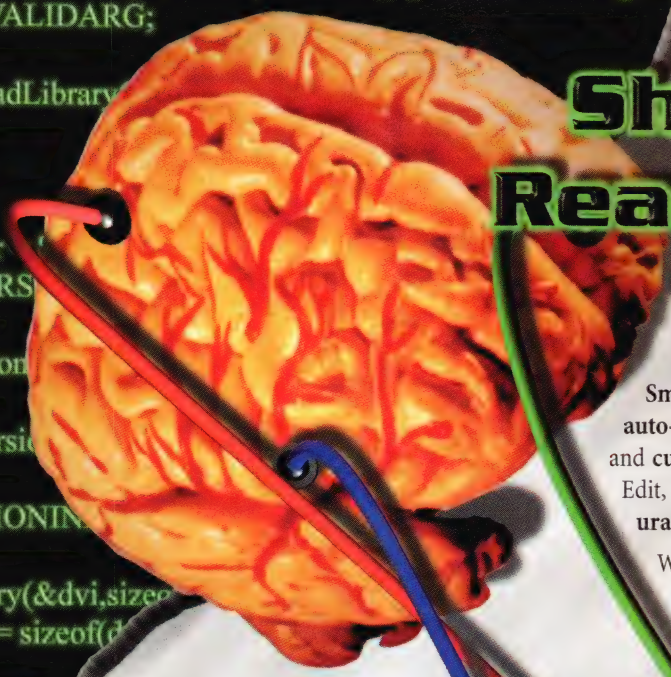
ROM functionality includes 4- and 1-digit decimal number display with an optional selectable decimal point. This is how ROM and firmware display sensor values, program numbers, and time. There are also codes to set and unset individual graphics symbols such as the IR transfer indicator. legOS makes these capabilities accessible via symbolic constants and inline wrappers in rom/lcd.h.

(continued on page 73)

Figure 3: *Downloading code from a PC to the RCX via an infrared link.*

"Portable Ports!"
Introducing the New RocketPort™ Serial Hub Si
Industrial-Strength Universal Serial Concentrator
*Cables not included
Put COM Port Connections Anywhere in Your Plant!
From the inventors of the multiport serial concentrator comes the most advanced universal serial concentrator on the market today—the RocketPort Serial Hub Si.
Now you can connect 4 or 8 native serial COM ports to your server from *any* location that has Ethernet access, even in the building next door!
You can now mix and match different types of serial devices depending on your required configuration, because each port on the RocketPort Serial Hub Si can be individually configured to support RS-232, RS-422, or RS-485.
The RocketPort Serial Hub Si installs in minutes—just plug it into the Ethernet network, install the NS-Link™ driver software on any Windows NT server, and connect your serial devices.
It's the only true universal serial concentrator on the market today!
Visit our Web site for more information, and be sure to register to win our "Feature of the Week!"
COMTROL
Connect with RocketPort™
www.comtrol.com
800-926-6876
©1999 Comtrol Corporation, RocketPort is a trademark of Comtrol Corporation. All other trademarks used herein are the property of their respective trademark holders.
900 Long Lake Road
St. Paul, MN 55112
Fax: 651-631-8117
e-mail: info@comtrol.com

Your Editor Should Almost Read Your Mind!



Multi-Edit is the only editor you'll need because it is designed to work the way you work, with the languages and tools you use everyday.

Smart indenting, code templates, construct matching and auto-commenting work as you type, eliminating redundant tasks and **cutting your keystrokes in half**! Like everything else in Multi-Edit, the templates, indenting and comment styles are **fully configurable and expandable**!

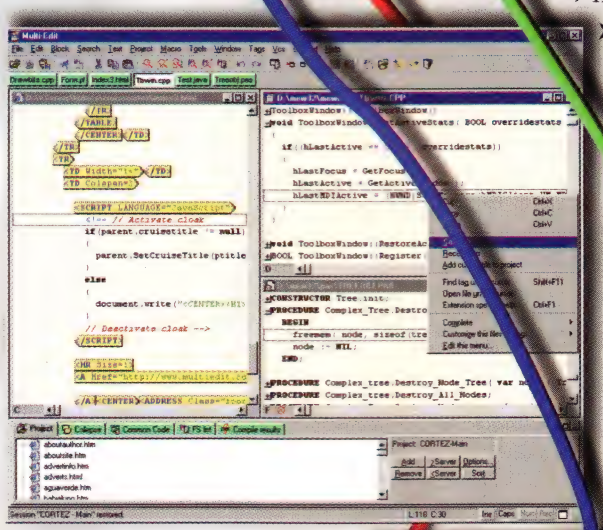
Whatever you code in, from C++, Delphi, COBOL & Fortran to HTML, Java, PERL or any of **30+ languages**, the support and integration you need is there for you. **RAD environment integration packages**, support for **dozens of compilers** and seamless **version control system integration** let Multi-Edit work directly with your existing development environment.

- Code collapsing and tagging
- Synchronized, split screen file compare & difference reporting
- Hex/binary editing
- Extensive find & replace tools
- Powerful extension language
- Project and session management
- Much more! Check out our website for details!

Programming for the Internet?

Built-in **WebLair web-editing and site management technology** and **integral FTP** give you the cutting edge tools you need to develop for the online world! Imagine having a complete, powerful HTML, ASP and embedded script editor all built into the same environment!

- Graphical HTML code highlighting, not just color coding.
- Edit embedded languages directly in your HTML code. Supports ASP, CSS, Java, JavaScript, PERL, PERLScript and VBScript.
- CommonCode allows you to easily manage, edit and update repeated HTML code segments. Much more powerful than server-side includes.
- Automatic previews in up to four different browsers simultaneously.
- Much more!



Try Multi-Edit Today, Free!

www.multiedit.com

The Editor that Thinks Like You Do

Sales: (800) 899-0100
Voice: (480) 968-1945
Fax: (480) 966-1654
sales@multiedit.com

Outside the US & Canada contact:
Soft/Export
www.softexport.com
info@softexport.com
Voice: +1 781 449 1440

multi edit

AMERICAN CYBERNETICS

Multi-Edit is a trademark of American Cybernetics, Inc. Multi-Edit is backed by an unconditional 30-day guarantee. Other products are the trademarks of their respective publishers.

Let Microway build your next Linux, UNIX or NT Workstation, Server or Beowulf Cluster using...

Screamer™ 667

Cray Performance at a PC Price!

667 MHz Alpha, 1.3 Gigaflops, 8MB Cache

Dual Alpha EV6 2000 Megaflops performance now shipping! Call Microway for a quotation.

Since 1982 Microway has provided the PC world with the fastest numeric devices and software available. No product in the last 17 years has excited us more than the Alpha Screamer. With its ability to execute 2.7 billion operations per second, the Screamer is the best choice for your next workstation or server! In addition to Linux and Beowulf applications, the Screamer runs NT, Digital UNIX and Open VMS.

This means you can run PVM and MPI Beowulf applications on the same hardware that runs Microsoft Excel or Word, Oracle, Fortran, C/C++ and Visual Basic. Also supported are Pro/Engineer, MicroStation, ANSYS, LAPACK, Gaussian, Softimage and Lightwave. Over the last 17 years we have designed systems for thousands of satisfied customers worldwide, including major universities, Fidelity Investments, Lucent Technologies and NASA. Our technicians are expert at configuring all Alpha operating systems, and you will not find more technically competent sales people anywhere.

Custom Screamer Workstations

Microway systems include 1 to 4 CPUs with fast caches, up to 4GB of high-speed memory, 6 PCI slots, SCSI hard drives, 3D graphics cards, DLT drives and libraries, and RAID subsystems. Microway's exclusive 8MB SSRAM cache, fed by a 288-bit wide memory system, boosts performance by up to 100%. Screamer™ workstations range in price from $2,795 to over $80,000.

Microway also produces one of the finest numeric optimized compilers - NDP Fortran. Since 1986, hundreds of applications have been ported to the x86 and Alpha with it. Using hand-coded BLA's and FFT's, our NDP VDSP Alpha Library hits 560 megaflops triangularizing dense arrays and performs a 1024 complex FFT in 146 microseconds. This library also includes all LAPACK subroutines.

AlphaPowered™

Press Accolades for Microway's Screamer

Windows NT Magazine Lab Report - August, 1998
"On the AIM WNT Peak Performance metric. . .Microway scored 643.7. It scored higher on the AIM WNT Sustained Performance metric-219.4 than any other Alpha NT System I've tested."

LINUX Journal - January, 1998
"Literally everything runs profoundly faster on the Screamer."

Windows Sources - February, 1998
"the Microway system blew away the best Intel-based workstations we've tested. . .on our number-crunching Lightwave 3D test."

PC Computing - July, 1997 ★★★★
Microway's Screamer . . . "is, quite simply, the fastest Windows NT machine on the planet. . . The performance leader."

Computers in Physics - September, 1997 *Product of the Month*

Visit **www.microway.com** for complete product information or call technical sales at **508-746-7341.**

Digital, Alpha, and Digital UNIX TM Digital.
Visual Basic, NT, Excel and Word TM Microsoft.
Screamer, NDP Fortran and Microway TM Microway.

Microway®

Technology You Can Count On

Corporate Headquarters: Research Park, Box 79, Kingston, MA 02364 USA • TEL 508-746-7341 • FAX 508-746-4678 • www.microway.com
info@microway.com • Australia 61 292094580 • Denmark 45 39624156 • Germany 49 6976752384 • India 91 806637770 • Italy 39 290782776
Japan 81 64593113 • Korea 82 25561257 • New Zealand 64 33595556 • Poland 48 22487172 • Spain 34 35809444 • UAE 97 19281081

Embedded Systems SHOWCASE

Paradigm C++ is a state-of-the-art integrated development environment that contains all the tools you need to develop and debug x86 embedded system applications. Paradigm C++ integrates an assembler, compiler, linker/locater, librarian, editor, debugger, version control, and more into a single development environment. Check it out for yourself and download an evaluation version of Paradigm C++.

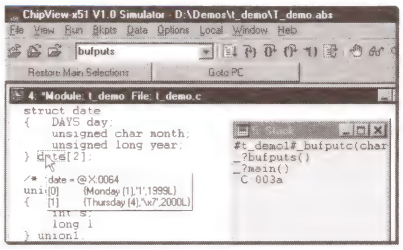
www.devtools.com/pcpp

Paradigm

Paradigm Systems
800-537-5043
info@devtools.com
www.devtools.com

Now with RTOS Awareness!

ChipView®-x51 to Turbo Debug 8051 C



- Supercharge your Nohau EMUL51-PC emulator
- Turbo Debug Power plus Windows ease-of-use
- Full Simulator supports on-chip peripherals, and interface to your own ActiveX, HDL or HTML pages!

Download a 30-Day Trial Today!
www.chiptools.com

ChipTools 905-274-6244 FAX 905-891-2715

CMX-RTX™ RTOS

Source Code Included • NO Royalties

80x51, 8051-XA, 80C251, 80x86, 80196/296, Z80/180, ST9, ST10, 80C16x, 68HC11/12/16, 68K, 683xx, H8/300H, H8S/2000, AVR, TLCS-900, ARM, M16C, SH, PowerPC, DSPs and More

Available NOW!

TCP/IP Network Layer

Compilers Simulators Debuggers

Available for most of the above processors Including 68HC05 & PIC16/17

CMX COMPANY
680 Worcester Road
Framingham, MA 01702 USA

Phone: (508) 872-7675
Fax: (508) 620-6828
e-mail: cmx@cmx.com
WWW: www.cmx.com

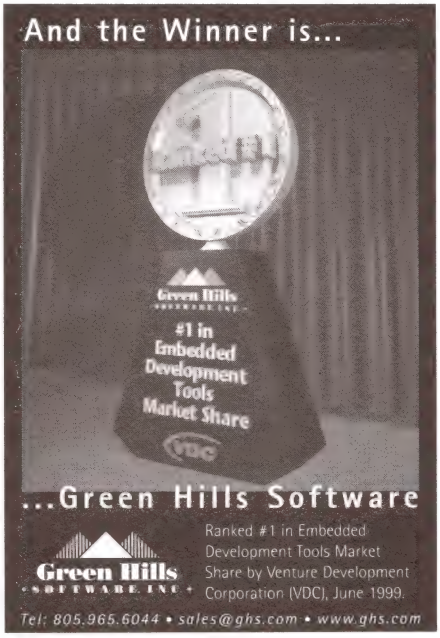
The Complete Package for Professional Embedded Systems Development Tools

AVOCET SYSTEMS®, INC.

- C Cross Compiler
- Macro Assembler
- Simulator Debugger
- RTOS
- In-Circuit Emulator
- Universal Programmer
- Extensive Chip Support
- Quality Tech Support

(207) 236-9055 (800) 448-8500
P.O. Box 490, Rockport, Maine 04856
sales@avocetsystems.com
www.avocetsystems.com

And the Winner is...



...Green Hills Software

Green Hills SOFTWARE INC.

Ranked #1 in Embedded Development Tools Market Share by Venture Development Corporation (VDC), June 1999.

Tel: 805.965.6044 • sales@ghs.com • www.ghs.com

FREE RTOS / GUI EVAL KIT

Includes:

- *smx* multitasking kernel
- *smx++* C++ API
- PEGmt multithreaded GUI for embedded systems
- WinBase to compile, debug, and run under Microsoft Developer Studio and Win32

www.smxinfo.com/spekdd.htm

MICRO DIGITAL INC
714-373-6862 800-366-2491

FREE DEMO

Want to improve your system requirements and architecture modeling and documentation?

AxiomSys is the leading PC-based tool for modeling systems requirements and architecture using the Hatley-Pirbhai real-time methods. AxiomSys validates the completeness of the model and provides a flexible report generator for complete documentation. Download the fully functional version of AxiomSys and tutorial at www.stgcase.com/ddj

STG

(800)959-2451
(661)296-3607
Fax: (661)296-5302
info@stgcase.com

Phar Lap Software, Inc.

Tel: (617) 661-1510
Fax: (617) 876-2972
www.pharlap.com
info@pharlap.com

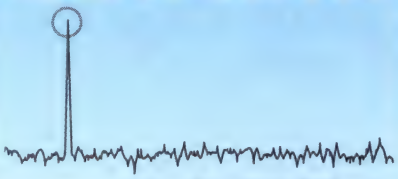
PHAR LAP

TNT Embedded ToolSuite

Realtime Operating System & Embedded Development Tools

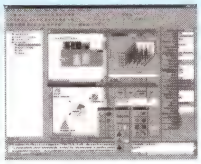
- RTOS supports a subset of the Win32 API
- Hard Realtime Deterministic OS
- Small mem. footprint: <100K
- Support for 32-bit x86 hardware
- Support for MS Visual C++ 6.0
- Support for the Microsoft IDE
- ETS TCP/IP – WinSock API
- GUI, MicroWeb Server, and more

Does your system skip a beat?



Don't wait to find out.

- RMA (Rate Monotonic Analysis)
- Timing simulation
- Network models
- RTOS support
- Friendly GUI
- Custom models



TimeSys Corporation
Real-Time... Real Solutions.

www.timesys.com
1-888-432-TIME

Embedded Systems SHOWCASE

ORBexpress™
Wicked Fast CORBA!

Don't waste another micro-second on the slow stuff! You can put your hands on ORB*express* performance and power today.

Using ORB*express* you can start building more adaptable real-time and embedded systems. ORB*express* is:

Lightweight – you can run ORB*express* embedded.

Flexible – you can build and plug-in your own custom transports.

Fast – you'll notice the difference.

Call Now! 1-800-800-6477
www.ois.com • info@ois.com

Objective Interface, the Objective Interface logo, and ORBexpress are trademarks of Objective Interface Systems, Inc. CORBA is a trademark of the Object Management Group. Copyright © 1999 Objective Interface Systems, Inc.

OBJECTIVE INTERFACE

WinDriver™
Automates Device Driver Creation & Cuts Development Time by up to 80%!

Same code runs on
Windows 95, 98, NT, 2000
Windows CE, LINUX

♦

No DDK or Kernel knowledge needed

♦

Free evalution copy
www.unipress.com/wddj

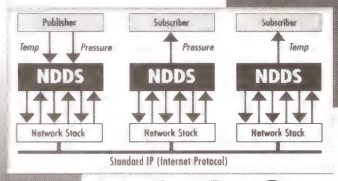
UniPress Software
800-222-0550 (USA) ♦ 732-287-2100

Accelerated Technology
INCORPORATED

NO ROYALTIES... SOURCE CODE... ANY CPU...
IT JUST MAKES SENSE

For more information:
www.atinucleus.com
phone: 1.800.468.6853
fax: 334.661.5788
email: info@atinucleus.com

Real-Time Networking Over Ethernet?



NDDS

The only middleware designed for real-time data distribution over Ethernet.

Ease your distributed, real-time application development with NDDS, the only middleware that provides fast, deterministic data distribution over Ethernet.

RTI Real-Time Innovations
REAL-TIME INNOVATIONS, INC.
www.rti.com

EYRX®
Embeddable RTOS from Eyring for 32 bit X86

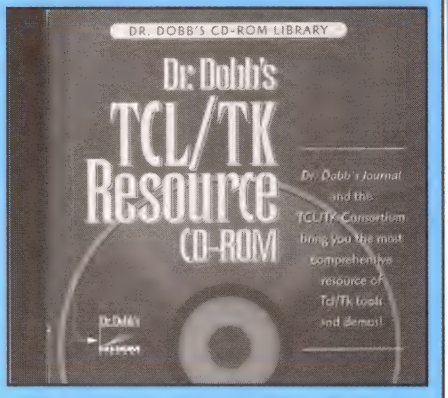
- VERY LOW COST ENTRY!
- Use Visual C++ development tools
- Protected mode, 4 Gigabyte mem. range
- Modular/scaleable, as little as 24k code
- DLL support
- Synchronous Message Passing
- Extensive POSIX compliance
- DOS compatible file system
- Boot from ROM, flash or disk
- Very fast switch times, flexible scheduler
- Configurable to any # tasks & priorities
- Embeddable TCP/IP

CALL FOR DEMO NOW!

EYRING CORPORATION
www.eyrx.com
800-921-5563
DISTRIBUTED BY T&T COMPUTER PRODUCTS, INC.

PRICE $24.95

Tcl/Tk Resource CD-ROM



Dr. Dobb's Journal & the Tcl/Tk Consortium deliver the most comprehensive resource of Tcl/Tk available on one CD-ROM! Tcl/Tk and its extensions are powerful tools for software development that allow developers to create scripts through a high-level programming language and graphical user interfaces with a toolkit that includes buttons, menus, listboxes, scrollbars and more!

Dr. Dobb's Tcl/Tk Resource CD-ROM is perfect for all programmers who need to get their projects out the door - fast! This CD-ROM contains programs and libraries that are ready to run, and easy to install.

PLATFORMS INCLUDED: AIX / Alpha OSF / Free BSD / HPUX / IRIX / Linux Macintosh / SCO / Solaris / Windows 95/NT / ... and more!

USE TCL/TK TO: Build web sites · Perform regression testing · Create new applications by integrating and extending existing components and applications · Act as an embedded command language · Shrink development time with this compact and efficient programming language

Call Now! 800-273-9208
Order from our web site! www.ddj.com/cdrom/

E-mail: orders@mfi.com
Fax: 785-841-2624

International: Use mail, fax, e-mail, or call 785-841-1631

Mail:
Dr. Dobb's CD-ROM Library
1601 West 23rd, Ste. 200
Lawrence, KS 66046-2703

Dr. Dobb's CD-ROM LIBRARY

(continued from page 70)

In-depth analysis of the ROM disassembly revealed further potential for enhancements. All ROM display functions employ the memory area 0xef43-0xef4b as a bit field indicating segment states. From this buffer, the display is updated by the *lcd_refresh()* function. Recognizing the rare possibility to improve both footprint and execution speed, I added direct-lcd.h to access the data area directly, bypassing ROM for all but display refresh (see Example 2 for a comparison). Gradually bypassing ROM completely is a general pattern in legOS development.

Notice how arguments to ROM functions need to be recopied. ROM calling conventions differ from gcc's—16-bit arguments and return values are passed in *r6*, not *r0*. Unfortunately, gcc's excellent support for assembler instructions with C operands only offers constraints for register classes, not individual registers (see Example 3 for an example of how button states are read). gcc also treats *r0..r3* as clobbered by function calls, although ROM preserves them. Trivial changes to the H8 machine definition file would have accomplished seamless compatibility with ROM, but I chose not to provide compiler patches for every gcc version or binaries for every platform.

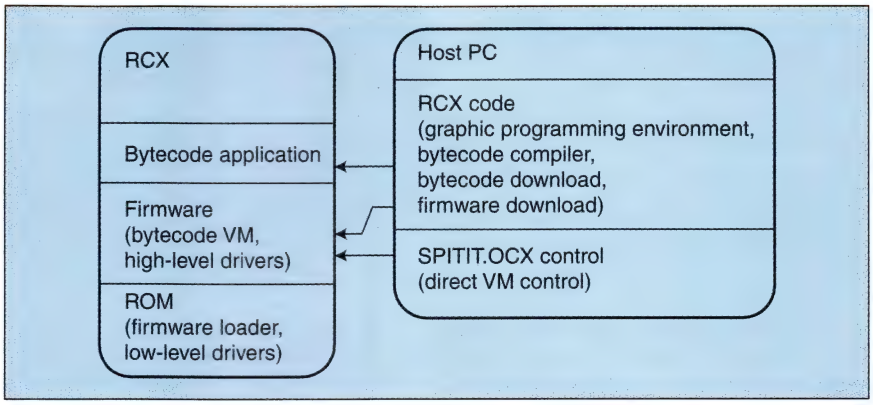


Figure 4: *RCX software layers.*

Direct access to the LCD bit field also expanded the display capabilities. Although the memory layout of every 7-segment digit is different, driver functions with a uniform interface could be implemented in conio.c with minimum footprint, due to the H8's powerful bit-manipulation functions. Providing softfonts to display hexadecimal or maimed ASCII characters paved the way to display hexwords and strings with *cputw(unsigned)* and *cputs(char*)*.

Memory Management

The choice of a memory-management algorithm was dictated by circumstances. Total system memory of slightly less than 32 KB and an overall design goal of small footprint ruled out page-oriented allocation schemes immediately; on average, they waste half a page per allocated block. Also, the smaller the blocks, the bigger their relative administrative overhead.

This left continuous allocation. Of the possible variants, I decided to use a straightforward scheme (see Table 1) that is optimal in terms of space allocation and linear with a low constant in the time domain. Why space optimal? For each allocated block, you need to store its length somewhere if you ever intend to free it.

"The compiler is really great and the support is the best I got from commercial software in my whole life..." - *Aschwin G.*

"GREAT, GREAT product. Keep up the good work!" - *Craig N.*

"Thanks for your great tools and service!!" - *Larry O.*

"And by the way, your case code is really nice. That is a genuine computer geek logic trick." - *Tom R.*

"This compiler has saved me a great deal of time and I like working with it." - *Gus M.*

"Thanks again for the excellent quality of your tools and technical support." - *Ocie M.*

HI-TECH Software has a range of ANSI C cross-compilers for most popular microcontrollers, including 8051, PIC, 6805, Z80, 68HC11, 68000, XA. Put some highlights in *your* programs - call now for more information or visit our web site. We have resellers in many countries.

HI-TECH Software
7830 Ellis Rd Ste 105
Melbourne FL
32904 USA

www.htsoft.com
sales@htsoft.com
Ph. 800 735 5715
Fax 407 722 2902

HI-TECH SOFTWARE

With a 32-KB address space and a processor in need of word alignment for all but character data, this translates to 14 bits. Furthermore, you need to track a block's status with a minimum of three categories—free, allocated, and reserved (for memory-mapped I/O areas). This would fit in the remaining two bits.

For multitasking systems, however, it is desirable to track block owners, if only to clean up after task death. With the constraint of word alignment, this translates to another two bytes of administrative data. Free and reserved blocks are efficiently implemented as special owner PIDs 0x0000 and 0xffff, respectively.

Address	What is Stored
First Block	
0xnnnn	Block length in words
0xnnnn+2	Owner's PID
0xnnnn+4	Data block
...	
Second block (mmmm=nnnn+4+2*length)	
0xmmmm	Block length in words
0xmmmm+2	Owner's PID
0xmmmm+4	Data block

Table 1: *Memory allocation scheme.*

To minimize fragmentation, freed blocks must be united with neighboring free blocks. This, however, means traversing the block list from the beginning to locate the predecessor. *malloc()* performs this traversal in any case, so unification was integrated there, leaving *free()* an atomic O(1) operation. Since the priority scheduler has to free process data and stack areas, this property is handy.

The memory-managed area starts immediately after the last byte of static data. Fortunately, the GNU linker provides a method of inserting symbols at precise locations in the memory map (see Example 4). This spares you from clumsy double compilation or compile-and-patch approaches. As a final common optimization, a pointer to the first free block is maintained.

Task Management

My next goal was preemptive multitasking. The subgoals involved in changing to a new task context can be clearly separated in a task switcher (which performs the actual context switch) and task scheduler (which decides where to spend the next timeslice). The latter should ideally be totally platform independent, whereas the former is machine-specific by definition.

To keep the scheduler interface lean and mean, I saved all context information on the stack frame. The only argument to the scheduler is the current task's stack address, and the only return value is the next task's. Consequently, the switcher is ignorant of process data structures. Starting a new task became the question of assembling a stack from which the switcher would return to the desired start address and having the scheduler return this stack address.

Initially, I selected a barebones circular scheduling scheme for maximum stability. The harder part was getting the switcher to work. I started with a cooperative approach, forcing tasks to yield explicitly. This gave me a semistable environment wherein I could control the exact moment a task switch would occur.

Trying to stick with UNIX standards, I examined various methods of task startup and shutdown. A fork seemed infeasible, as it requires a full copy of both data and stack segments. It is also overkill for most embedded applications. Finally, I extended the *exec* family with *execi()*. This function inserts a new process into the scheduler and allocates stack space for it. *execi()* is nonstandard in that it doesn't replace the old process, but simply adds a new one. In more orderly fashion, *exit()* ends a process' life, and *kill()* allows it to prey on others.

After this system proved reasonably stable, I plunged into the joys of IRQ handling. As Table 2 shows, Hitachi placed the H8's IRQ vector table at 0x0000 as usual—a ROM address. However, in accord with the scheme of downloadable firmware, Lego's engineers displayed some excellent software engineering. They created a secondary IRQ table in RAM and a primary ROM wrapper for every IRQ (see Example 5). The secondary IRQ handlers can be easily overridden by new firmware. Replacement handlers have to return with a normal "subroutine return" instruction, and they can rely on *r6* being saved and restored by the wrapper.

The H8/3929 series provides one 16-bit and two 8-bit on-chip timers, each with several associated interrupts for compare-to-constants and overflow events. Fractions of the system clock are available as internal clock frequencies, as well as various external trigger modes. The 16-bit timer can generate two PCM waveforms, the 8-bit timers provide one

Address	Memory	Function
0x0000–0x0049	On-chip ROM	IRQ vector table
0x004A–0x3FFF	On-chip ROM	
0x4000–0x7FFF	Reserved	
0x8000	External RAM	Firmware entry point
0x8001–0xEFFF	External RAM	
0xF000	External register	Motor controllers
0xF001–0xFB7F	External RAM	
0xFB80–0xFD7F	Reserved	
0xFD80–0xFF7F	On-chip RAM	
0xFF80–0xFFFF	On-chip register field	

Table 2: *Hardware memory layout.*

```
(a)
> cd binutils-2.9.1
> ./configure --prefix=3D/usr/local --target=3Dh8300-hitachi-hms
> make
> su
# make install

(b)
> cd gcc-2.8.1
> ./configure --prefix=3D/usr/local --target=3Dh8300-hitachi-hms
> make LANGUAGES=3Dc
> su
# make LANGUAGES=3Dc install
```

Example 1: *Cross-compiler setup. (a) Compiling and installing binutils; (b) compiling and installing gcc.*

```
dlcd_show(LCD_S1_SELECT);
9000:    79 03 ef 49        mov.w   #0xef49,r3
9004:    7d 30 70 00        bset    #0x0,@r3

lcd_show(s1_select);
9000:    79 03 30 08        mov.w   #0x3008,r3
9004:    6d f6              mov.w   r6,@-e7
9006:    0d 36              mov.w   r3,r6
9008:    5e 00 1b 62        jsr     @0x1b62:0
900c:    6d 76              mov.w   @r7+,r6
```

Example 2: *Direct LCD manipulation versus ROM calls.*

Jbed

Pure Java™ based RTOS for embedded systems

Be fast
Jbed runs faster than other Javas because it always compiles, never interprets.

Stay small
Jbed's small memory footprint (starting from 8 kBytes) allows for low-cost hardware with less memory.

Save time and cost
Jbed offers the productivity advantage of the Java language to the embedded world.

Features included
- *Integrated Development Environment*
- *Deadline-driven, time-triggered scheduling with admission testing*
- *Incremental Garbage Collector compliant with hard real-time*
- *Reflection, serialization*
- *All standard communication protocols supported*
- *Hot-loading with esmertec's TBCC (target byte code compiler)*
- *Y2K compliant*

Jbed

Hard RTOS and JVM in one!

esmertec inc., Technoparkstrasse 1, CH-8005 Zürich, Switzerland,
www.esmertec.com, info@esmertec.com, F +41 1 445 37 30, X +41 1 445 37 34

esmertec

your partner-company for embedded real-time Java

```
extern inline int button_state(void) {
     int rv;
__asm__ __volatile__(
     "mov.b @@xb7:8,%@1\n"
     "mov.b @@xbe:8,%@h\n"
     : "=r" (rv)                          // output    (general register)
     :                                    // input     (none)
     : "cc"                               // clobbered (condition codes)
     );
     return rv;
}
```

Example 3: *Assembler instructions with C operands.*

```
bss : {
         _bss_start = . ;
        *(.bss)
        *(COMMON)
         _end = . ;
         _mm_start = . ;            /* start memory management here */
      }  >ram
```

Example 4: *Inserting symbols in the linker map.*

```
              46c:      6d f6          mov.w r6,@-e7
              46e:      6b @6 fd 92    mov.w @@xfd92:16,r6
              472:      5d 6@          jsr   @r6
              474:      6d 76          mov.w @r7+,r6
              476:      56 7@          rte
```

Example 5: *Disassembly of a ROM IRQ wrapper.*

(continued from page 74)
each. Additionally, an on-chip watchdog timer is available.

Experimentation with the timers quickly indicated that the first 8-bit timer's output was wired to the speaker, ruling it out for task switching. As the second 8-bit timer appeared to be safe, I employed it for my first scheduler. Later, I discovered it was generating the IR carrier frequency, so I moved the task switcher to the 16-bit timer.

It was somewhat disappointing to notice that some IRQ handlers may crash the system. For IRQs associated with certain status bit changes, the handlers have to reset the status bit before returning or the system will freeze. You should always mask out IRQs instead of providing empty handlers. The otherwise comprehensive, freely available hardware manual from Hitachi (link on http://www.multimania .com/legos/) fails to address this fact.

The current scheduling algorithm is priority based. Per default, only pairwise different priorities are admitted, as this allows for considerably faster scheduling. Support for equal priority levels by priority rotation within a level can be compiled in.

Processes may wait for arbitrary event functions to become nonzero — this is how *sleep()* is, and *select()* may be implemented. To conserve battery power, I introduced a special idle task of lowest priority. It switches the processor to sleep mode until the next interrupt occurs in case all regular processes are waiting for events.

Additional Features

The combination of memory and task management prompted a means of synchronizing access to critical sections. POSIX semaphores are an accepted solution to this problem, and the POSIX standardized interface was easy to implement.

Writing an IR driver was frustrating until I had a flash of inspiration. No matter what settings I tried for the H8's on-chip serial module, the PC wouldn't receive anything. Then I realized the carrier frequency had to be generated on-chip. The IR driver supports half-duplex operation at arbitrary baud rates and collision detection. Hardware constraints limit usable transmission speeds to 4800 baud, but this allows over four-times faster communication than the original firmware for limited distances. For the near future, there is talk of writing packet networking code.

About that time, another programmer discovered the motor driver I/O address to be 0xf000. That same day, I wrote a first PCM driver. A later discussion on the Mindstorms mailing list led to an advanced PCM driver derived from the Bresenham line-drawing algorithm. It offers 256 speed levels instead of the eight available with the standard firmware.

Support for basic resistive sensors (that is, the touch and light sensors packaged with the Mindstorms set) was straightforward to add with code taken almost literally from the H8/3292 hardware manual. Guessing conservatively, a 10-KHz sampling rate can be achieved.

Making the light sensor measure reflectivity instead of just ambient light was largely trial-and-error. Power for the LED had to come from some specific output pin, and there were only a handful to try. As the cable to the sensor has only two wires, alimentation and A/D conversion are mutually exclusive. They have to be synchronized. Rotation sensors additionally require a state machine to decode.

Examinations of the RCX hardware indicate that two additional A/D pins might be usable. Should this be confirmed, legOS will support two additional sensors.

Conclusion

There's plenty of room for improvement in legOS. Apart from C++ support, device files, and enhanced stability, most items on the wish list depend on IR packet networking.

A shell task could provide a clean method of loading applications via IR as previously outlined, saving application developers from downloading the kernel every time. Together with enhanced IR speed, this could potentially divide program download times by four or more.

A gateway application on the host computer could provide TCP/IP connectivity, enabling services such as local and remote joystick control and status display. Commercial products such as EMIT 3.0 (http://www.emit.com/), available with an H8 evaluation board, might serve as an inspiration.

Packet networking would also vastly extend the possibilities of inter-RCX communication, facilitating the development of systems of interacting agents.

legOS already fills a technically interesting niche for development targeting the Lego Mindstorms. With some of the aforementioned extensions, it might become a feasible platform for courses in introductory robotics, mobile agent systems, and certain forms of rapid prototyping.

DDJ

In JAVA, Speed is a BIG Advantage.
You've thoroughly optimized your Java code for performance. Don't let your database engine slow you down.
For the best in small footprint, flexible, portable and fast database technology, depend on Raima's *Velocis Database Server* with its new, pure, fast-loading JDBC driver.
Raima Corporation's Velocis DBMS is Built for Speed.
Download the Java-Enabled Velocis™ from:
www.raima.com/java
or call 1-800-327-2462
or 206-515-9477
If it's fast – it's
RAIMA™

JBED: Java for Real-Time Systems

A JVM and an RTOS

Jörgen Tryggvesson, Torbjörn Mattsson, and Hansruedi Heeb

Jbed is a small, fast Java Virtual Machine (JVM) for embedded real-time systems. It also includes a complete real-time operating system (RTOS). In designing Jbed, our first goal at Esmertec (http://www.esmertec.com/) was to make a seamless combination of an RTOS and JVM without any glue code in between. Our second design goal was not to rely on an interpreter or just-in-time (JIT) compilation. Instead, Jbed uses either a precompiled native image, or a built-in compiler that translates Java bytecode into native code at load time. (This technique is usually referred to as "flash compilation," but since someone trademarked that term, we call it "TBCC," short for "target bytecode compilation.")

Because Jbed always runs natively, it is a very fast Java system. That it is also a small system may be surprising, but leaving out the interpreter and not relying on techniques such as JIT saves memory. Additional memory saving is achieved by writing everything in Java—even the device drivers (to write device drivers in Java, Jbed provides a built-in class with methods for direct memory access). Of course, you can link to legacy code written in C, but if you refrain from that, only the Java memory management has to be loaded, and a lot of overhead is avoided. A complete Jbed system with TBCC is only about 256 KB and uses about the same amount of RAM to run. A system with TCP/IP and a web server is about 128 KB. Minimal precompiled systems without the TBCC can be smaller than 10 KB. These numbers include both the RTOS and JVM.

TBCC technology has another benefit. Code loaded dynamically into the running system is compiled immediately, then executed at full native speed. This makes Jbed ideal for applications where interpreters are too slow, but where hot-loads of patches or new applications are necessary (as with some Jini applications, for instance).

A final design goal was to make Jbed suitable for hard real-time applications. Hard real-time usually means that no outside event must ever be handled too late. More precisely, each event has a given

deadline and missing any such deadline results in an error condition (soft real-time, on the other hand, allows for occasional deadline misses). Real-time applications are more challenging under a Java-based operating system than under other systems because Java uses garbage collection. The garbage collector used in Sun's original JVM cannot be interrupted and is therefore unsuitable for real-time applications. Instead, an incremental garbage collection algorithm has to be used that can be interrupted anytime.

Jbed is a clean-room design and contains no Sun sources. It comes with its own integrated development environment (IDE), or can be used with Symantec's Visual Café, IBM's VisualAge, and the like. Jbed runs on Motorola 68xxx and PowerPC processors, with Intel, ARM, and MIPS processors support coming soon.

Hard Real Time

Contrary to popular notions, hard real-time problems with extensive hardware interaction can be solved entirely in Java. To illustrate this, we present a hard real-time example that is simple and maintainable, while still meeting hard real-time design requirements. In doing so, we won't link C code for the device drivers, but instead use the hardware interface libraries that come with Jbed. This not only makes memory management easier, but has advantages in terms of documentation and code maintenance. Because the problem is hard real time, Java systems that are interpreted or rely on techniques such as JIT compilation can't be used.

The problem at hand involves a robot with two moving axles, both under control of the system; see Figure 1. The robot has an attached laser-pointer that is used to display Lissajous figures on a screen (a Lissajous figure is the superposition of the two sine functions $x(t)=A\sin(ct)$ and $y(t)=B\sin(dt+e)$ plotted in the x-y-plane; see Figure 2. This example is hard real time because the movements of the robot have to be monitored and adjusted in very short time intervals (milliseconds) to achieve the desired accuracy. The accuracy of the system can be easily tested by displaying a Lissajous figure where the two sines have the same frequency and no phase difference because the resulting figure must be a line. The slightest inaccuracy of the timing makes it into an ellipse instead. Finally, the robot system will be implemented as a time-triggered (deadline-driven) system.

Event-Triggered versus Time-Triggered Systems

In event-triggered systems (based on interrupts) that use multiple sensors (heat sensors, oil-pressure sensors, sensors triggered

Jörgen is development engineer at Mecel AB. He can be reached at jorgen.tryggvesson@mecel.se. Torbjörn is a software consultant working at Mecel AB and can be contacted at info@mecel.se. Hansruedi is CEO of Esmertec Inc., and can be contacted at hh@esmertec.com.

Some lead, others follow.

Tornado II. The new pinnacle in embedded development.

WindRiver® SYSTEMS

An ISO 9001 Registered Company

Consumer Electronics	Imaging	Data Communications
Telecommunications	Intelligent I/O	Medical Electronics
Military/Aerospace	Transportation	Industrial Control

Whatever your market, we have your embedded software.

TORNADO II

Wind River Systems has always been a trailblazer in the embedded market. Tornado gave developers, for the first time, a turnkey solution that was integrated, intuitive, completely open and extensible, and tailored to the needs of complex, fast-turnaround projects. Well, a new path to the future has just been cleared. Tornado II is a quantum leap forward—offering unprecedented ease of use and best-of-class technology for embedded graphics, code visualization and more.

Now embedded developers have an even more powerful tool. And, coincidentally, our competitors have an even higher peak to climb. For more information on Tornado II, the new pinnacle in embedded development, call 1-800-545-WIND or visit our Web site at www.wrs.com.

© 1998 Wind River Systems. Wind River Systems and the Wind River logo are registered trademarks and Tornado is a trademark of Wind River Systems.

(continued from page 78)
once per rotation of a moving part, etc.), each trigger is an interrupt handled by an interrupt routine. The interrupt routines are short, and as much work as possible is delegated to nonreal-time threads. (Here, we use the term "thread" exclusively for Java style, nonreal-time threads, and we use "task" for real-time tasks.) Each interrupt routine has a priority. Routines with higher priority will interrupt routines with lower priority.

In such a system, priorities have to be assigned (possibly dynamically) such that each external event is serviced within its deadline. For instance, the interrupt routine that must be called once per rotation of a moving part must have a priority that guarantees it is still called once per rotation even under special circumstances (such as a malfunction of another part of the machine). If this is not guaranteed, a malfunction could spread quickly. In a complex system, this priority assignment can be very difficult to debug.

In time-triggered (deadline-driven) systems, on the other hand, all sensors are sampled (polled) at regular intervals. Each sensor causes the execution of a small piece of code, similar to an interrupt routine. Again, as much of the calculation as possible is delegated to nonreal-time threads. Instead of priorities, the routines triggered by the sensors have deadlines that are usually known *a priori* and need not be assigned. For example, a rotating part that rotates a maximum of 1000 times per second has a natural deadline of 1 millisecond. The scheduling is based on which deadline is the most urgent (earliest deadline gets first scheduling).

Because the routines usually do not perform complex calculations, it is possible to assign them a duration (that is, a maximum execution time). Again, the duration is known *a priori* and, therefore, the scheduler can do a load-time analysis of the schedule of the system under all circumstances. After all modules are loaded (either statically and one-time or dynamically), Jbed guarantees that the system can be scheduled under all circumstances if the durations of the real-time tasks were given correctly. If the system cannot be scheduled, it does not even start executing. The requirement to have a correct estimate of the duration of the real-time tasks is not a serious problem in practice, because well-designed tasks are simple and the estimates can be on the safe side (and the actual durations are given in the Jbed debugger). In our robot example, it is guaranteed that the service routine for all moving parts is executed in time under all circumstances.

How does the responsiveness of event-triggered systems compare to their time-triggered counterparts? Under small loads (few external events), event-triggered systems are more responsive because each event is serviced immediately (unless a higher priority event comes in between). Time-triggered systems have a response time given by the sampling period. Under high loads, however, time-triggered systems excel because they have much less overhead than event-triggered systems.

Time-triggered systems have another advantage. If the machine in our robot example is extended by another rotating part, it is easier to fit the new control software into the system if it is time triggered. The service routine of the new part has a known deadline and duration, and can therefore be added to the system without changing the existing service routines. If the system can still be scheduled, no further work has to be done (if it is not, Jbed detects this at load time and the system will not start up). Event-triggered systems,

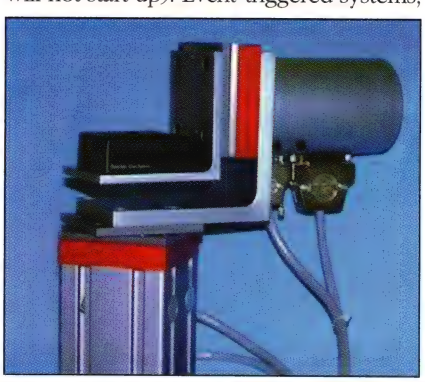


Figure 1: *The robot built by the Institute of Robotics, ETH, Zürich.*

Win32 RTOS for x86 Embedded Systems

On Time's embedded systems RTOS implements a Windows NT subset kernel in only 16k of memory.

On Time's 32-bit RTOS has a scalable component architecture. Only the components actually needed by the application are loaded on the target.

RTOS Features:

- Source and binary compatibility with Windows 95/98/NT/2000
- Supports DLLs
- Supports Borland, Microsoft, Watcom, Delphi Compilers
- Unmatched real-time performance
- Context switch 0.73µs on P120
- Supports (but does not require) PC compatible hardware
- Boots from disk, BIOS extension, ROM, or DOS
- Free tech support (less than 24 hours turnaround from developer team guaranteed)
- Full source code available
- **Free Evaluation Kit** available at http://www.on-time.com
- **No run-time royalties**

RTOS Components:

RTTarget-32
Core Operating System and Development Tools
Price: $1700

RTKernel-32
Real-Time Multitasking Kernel for RTTarget-32
Price: $1950

RTFiles-32
FAT File System for RTTarget-32
Price: $1700

RTIP
TCP/IP Stack for RTTarget-32
Price: $7500

x86 Embedded Systems at Prices that Won't Make You WinCE

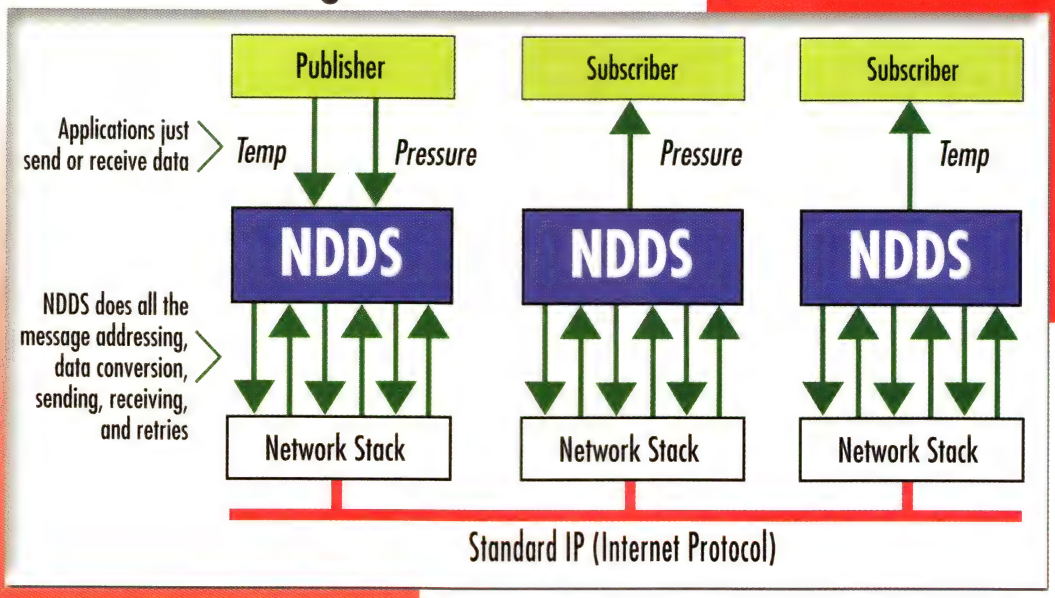
North America: On Time • 88 Christian Avenue • Setauket, NY 11733 • USA
Phone (516) 689-6654 • Fax (516) 689-1172 • email info@on-time.com

International: On Time • Hofweg 49 • 22085 Hamburg • Germany
Phone +49-40-2279405 • Fax +49-40-2279263 • email info@on-time.de

http://www.on-time.com

On Time
REAL-TIME AND SYSTEM SOFTWARE

Real-Time Networking Over Ethernet?



Networking Made Simple™

N D D S

The only middleware designed for real-time data distribution over Ethernet.

Writing network code is tough for any programmer, tougher still to program complex, time-critical data flows.

NDDS simplifies distributed, real-time application development.

- The NDDS publish-subscribe architecture is a simple, natural way to design many-to-many data communications.
- The NDDS real-time publish-subscribe extensions distribute data quickly, efficiently, and deterministically.

Ease your distributed, real-time application development with NDDS, the only middleware that provides fast, deterministic data distribution over Ethernet.

For more information on NDDS and fast, deterministic data distribution over Ethernet, see the white paper, *Can Ethernet Be Real Time?* on our web site.

RTI
Real-Time Innovations

www.rti.com

REAL-TIME INNOVATIONS, INC.

155A Moffett Park Drive • Sunnyvale CA 94089 • (408) 720-8312 • (408) 734-5009 fax • info@rti.com • www.rti.com

©1998 Real-Time Innovations, Inc. All rights reserved. NDDS and Networking Made Simple are trademarks of Real-Time Innovations, Inc.

(continued from page 80)
however, will likely require many changes. Time-triggered systems, thus, pave the road for the use of software components (like JavaBeans) in real-time systems.

Java Library for Real Time

Jbed does not make any changes to the Java language. To support real-time embedded systems, Jbed adds library classes — some of which are built in (to access hardware directly from Java, for instance), while others are regular libraries written in Java. The most important such library is the task library, com.jbed.tasks.

The task library defines a class *Task*, which is a subclass of *java.lang.thread*. Real-time tasks in Jbed therefore behave exactly like Java threads, except that they give additional guarantees concerning their real-time behavior. As with Java threads, *Task* objects control the behavior of a concurrent activity; they contain all the bookkeeping information necessary to do this. You implement the activity itself by writing a class that implements the interface *java.lang.Runnable* — again, just as with regular Java. The *run* method of this interface contains code that makes up the concurrent activity. As in Java, an alternative to this approach is to subclass the *Task* class and implement user code in the *run* method of the class. In typical nonreal-time programs, a thread's *run* method contains an endless loop. This is different for real-time tasks: A *Task* should be restarted by Jbed so that it can automatically start the next iteration at the right time.

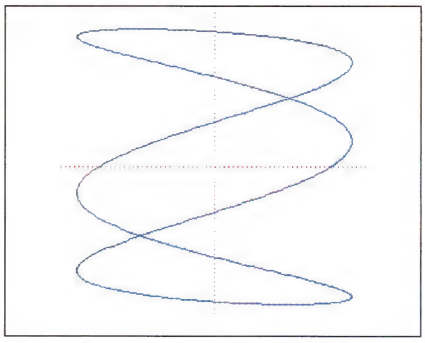


Figure 2: *A Lissajous curve.*

The Jbed scheduler uses timeslices to assign computation time to tasks. A timeslice is an implementation-dependent short period of time; one millisecond, for instance. The implementation's timeslice is provided in the static variable *com.jbed.tasks.Task.timeslice*; see Listing One. The scheduler installs a timer interrupt to get control (after a timeslice has passed) by preempting the currently running task. The scheduler then checks whether the previously running task has exceeded its admitted duration; if so, it is aborted. The scheduler determines which task has the closest deadline to meet, and lets it run for the next timeslice. Rescheduling may occur because of preemption after a timeslice, and also voluntarily when the running task stops or blocks (for instance, by sleeping or waiting for a signal).

Jbed distinguishes between the six types of real-time tasks in Table 1. The type of *Task* is given in the constructor using the *RealtimeEvent* argument.

When a task is created, its constructor initializes it if the scheduler's admission testing was successful. This is the case if the scheduler can provide sufficient computation time to the new task; that is, if other tasks have not yet reserved too much of the processor's capacity. Otherwise, a *com.jbed.tasks.AdmissionFailure* exception is thrown. Listing Two is a typical example of how a task is created. The first statement creates an instance of one of the real-time event classes. Depending on the chosen event class, the real-time behavior of the task is different. Depending on the event class, you may need to use a *try/catch* statement for creating the event object. The second statement creates an instance of a runnable object (an instance of a class implementing the *java.lang.Runnable* interface). This object implements your application-specific behavior. The *try/catch* statements create the

TRECK REAL-TIME EMBEDDED INTERNET PROTOCOLS

We know that schedules are always tight, and that you do not have time to "port" Internet protocols into your embedded environment (let alone write them from scratch). That does not stop marketing people inside your company from asking for Internet functionality in your new product line that is rolling out in the next few weeks. Now you have someone to turn to...

- **The FASTEST TCP/IP for Embedded Systems**
- **Fully TESTED and Qualified**
- **SMALL Code Footprint**
- **RFC2000 Compliant**
- **Processor, RTOS, and Compiler Independent**
- **Written from "scratch" for Embedded Systems**
- **BSD4.4 Sockets Interface**
- **Base TCP/IP includes TCP, UDP, ICMP, IP, ARP, ETHERNET, SLIP, PPP, and PING**
- **Optional Products include SNMPv1, SNMPv2, SNMPv2c, SNMPv3, WebServer, POP3, SMTP, PPP, BootP, DHCP, FTP, and Telnet**
- **NO PORTING REQUIRED**

Treck Inc.

(800)340-6648 or (513)688-0553
or visit our web page at www.treck.com

USB On Windows® NT® 4.0 and NT Embedded 4.0

USB Extension Available to WinDK™

- Same USB driver source for Windows NT Embedded 4.0, Windows NT 4.0, Windows 2000 and Windows 98.
- Hot Pluggable support added for Windows NT Embedded 4.0 and Windows NT 4.0
- Full support for all USB transfer types.
- More USB samples, the kind you really need!

Unmatched Support

From some of the world's most respected device driver engineers: we've supported USB for over 3 years.

Systems Integration, Consulting & Training Too!

Our customers agree and include Microsoft, IBM, Intel, Matrox, Epson and many others.

Tools, Training and Support From A Team You Can Work With

BlueWater Systems Inc.

800.962.2114 • 425.771.3610

www.bluewatersystems.com • sales@bluewatersystems.com

task object, using the event object and the runnable object as parameters, plus the duration, allowance, and deadline values specified in microseconds. The last statement starts the task, which means that it is marked as ready to run. The scheduler then executes the task at the appropriate time.

Two arguments to the *Task* constructor, *duration* and *deadline,* have been explained already. The additional argument *allowance* requires further explanation: It is the time required for the handler of the *DurationOverflow* exception that is thrown if the task exceeds its duration. Without an allowance for this, the system might not have any time left to schedule it. Frequently, this situation is simply an error condition that requires no handling, and is removed during the debugging phase. Then the allowance can be set to zero. The task's constructor throws an *IllegalArgumentException* exception unless the following holds:

```
duration > 0
allowance >= 0
deadline > duration + allowance
```

If a system contains both Java threads and real-time tasks, Jbed schedules the threads whenever no task is running according to its priority (round-robin if several have the same priority). An example of such a thread is the garbage collector.

The Robot Example

Recall that the goal in our real-time robot example is to move two orthogonal robot axles such that their combined movement follows a Lissajous figure with given parameters. The example has three real-time tasks and no user-defined Java threads. One of the three tasks, the *WatchDog,* is simply used to check if the software is alive and running. If it is not, the hardware shuts down. The other two are the *Planner* task and *Control* task.

To reduce the memory footprint, the example avoids repeated dynamic memory allocation and the target bytecode compiler (TBCC) is not used. Jbed is modular, and unused modules (such as the garbage collection or the TBCC in this case) are not loaded. Dynamic memory allocation is avoided by using static variables. Listing Three presents the definitions of some of these variables and definitions (the complete listings are available electronically; see "Resource Center," page 5).

The *Planner* task simply calculates the desired position of the two axles from the given parameters at the current time and then advances time. The *sine* method (from Jbed's math library com.jbed.math .FMath) is used instead of the Java math library, which is far too large for embedded systems; see Listing Four.

The *Control* task has the more complex job of comparing the actual and the desired positions and calculating a reasonable movement for the current timeslice. The classes *com.jbed.peripherals.Counter* and *com.jbed.peripherals.AnalogActuator* interface to the actual hardware and allow mapping of, for example, an analog input reading from hardware to a given floating-point range. In our robot example, a mapping to an angle in radians was chosen in the initialization; see Listing Five.

So far we have defined what the real-time tasks do. Now we need to set them up such that they run in real time within reasonable time constraints. Since this robot example is time triggered, the tasks are to run at regular intervals and poll the hardware. First, a *PeriodicTimer* event is set up that fires every 5 milliseconds. Then a periodic *Task* is created based on that. The *Task* has a duration of 2 milliseconds—no allowance for an error handler and a deadline, which is the same as the period; see Listing Six.

Because the *Planner* does trigonometric calculations, it runs at twice the period of the *Control* task, namely 10 milliseconds. To do this cleanly, Jbed offers a special type of task called "harmonic" task. The corresponding *RealtimeEvent* is a *HarmonicEvent* that

C DEVELOPMENT TOOLS

Quicktionary SDK

from MetaWare

NEW!

Develop applications for Wizcom's new QuicktionaryII handheld processor with MetaWare's rock-solid tools.

- The QuicktionaryII scans and recognizes text and barcodes.
- Use its ARM7 processor to develop your own applications:

  inventory applications
  order entry
  library check-in and check-out
  and whatever fits your vision

See the Quicktionary at Booth S1272 at Comdex, November 15-19!

MetaWare's tools include the famous High C compiler toolchain customized for Thumb development on Quicktionary, along with the SeeCode debugger.

www.quicktionary.com
www.metaware.com

MetaWare also provides professional-quality C/C++ development tools for ARC, ARM (including StrongARM), picoJava, and PowerPC targets.

MetaWare

2161 Delaware Avenue
Santa Cruz, CA 95060-5706
phone 831.429.6382 (META)
fax 831.429.9273 (WARE)
email techsales@metaware.com

(continued from page 83)
triggers at every second start of the *Controller*. With these six statements, you have set up the entire real-time system.

The ADAPT Project

The Advanced Diagnosis with the Applet Technology (ADAPT) project is a European Union-supported collaboration between Mecel (a Swedish company that develops electronics and software for the automotive industry; http://www.mecel.se/) and Trialog (a French software house that focuses on embedded systems and home/vehicle connectivity solutions; http://www.trialog.com/) that aims to develop methods for remote downloading of software to cars.

As cars become more and more complex, much of their functionality is controlled by electronic control units (ECUs). These ECUs are connected and communicate with each other in networks. When an error occurs somewhere in the network, the symptom can occur at a totally different place. Thus, it is hard to localize faults in new cars and that is why a good diagnostics system has become more and more important. The main point is to be able to download and run Java-based diagnosis components (Diaglets) via, for example, a diagnostic web site.

To enable this, at least one of the ECUs in the car has to support code downloading. In the ADAPT project, a telematics unit (TU), with GSM modem, is used inside the car as a communication link between the diagnostics web site and car. The TU uses Jbed as the operating system, which has a built-in Java virtual machine (JVM), and code can dynamically be loaded and unloaded via GSM. It will also be possible to download diaglets from the TU to other ECUs inside the car if those ECUs have JVMs. In the ADAPT project, all parts of a complete system for remote car diagnostics are taken care of—programs for the diagnostics web site, all embedded programming inside the cars, security and safety aspects, and so on. To simplify development, all programming is in Java.

Device Drivers in Java

One of the most fundamental qualities of the ADAPT system is dynamic downloading of code, which is a feature that the Java language in itself supports. That was one of the main reasons why we chose Java in the first place. After we decided to use Java, our next decision was whether to write the device drivers in C and use a native interface between C and Java, or write them in Java. C is often seen as the natural choice for drivers because it allows tight control of the hardware, but Java drivers have some advantages. The most obvious is that you have one consistent code base that can be manipulated with one set of tools, instead of having two different sets of code with two different sets of tools. There is also no need to have any awkward half-Java/half-C interfaces between the two.

Java has some other advantages compared to C. For example, javadoc is a built-in documentation system that produces documentation in HTML format. Of course, tools like this exist for C, but it is a standard tool in Java and you can count on it being there.

Also, Java produces cleaner code and less room for errors by having features such as strong typechecking, garbage collection, exceptions, and no pointer arithmetic.

Because of the safety checks Java makes during run time, Java is slower than C, even when compiled (of course, this depends on your compiler). Listing Seven shows some of the assembly code generated by the compiler for the innermost loop in one of our communication-bus-driver interrupts. This code is performance critical since failure to execute in time leads to loss of data. In Listing Seven, the extra lines of code added for safety checks are highlighted. These add 20–25 percent to the execution time. But the safety checks increase stability and reduce debug time, so they are a definite advantage for all but the most speed-critical code. In Jbed, you can turn off run-time checks on a per function basis. The possibility to turn them off in performance-critical parts of the code gives us the best of both worlds.

Another tricky area when writing device drivers involves interrupts. Jbed can handle interrupts in two different ways. The old-fashioned way has a pointer to the interrupt-handler function, which executes immediately and bypasses the normal scheduling of threads and tasks. (Jbed differentiates between normal Java threads, which execute with no real-time requirements, and real-time tasks that are guaranteed to execute within a deadline.) In the other scenario, Jbed can schedule the interrupt as a real-time task that cooperates with the other threads and tasks. Tasks, however, cannot be scheduled with greater granularity than the minimum timeslice—in our case, a 22-MHz 68376, which was set to 1 millisecond. Because the interrupts from the CAN bus could come as often as every 200 µs (at 500 Kbaud), we were forced to use the traditional approach.

Having interrupt handlers preempting the normal threads and tasks creates another set of problems:

- You cannot expect the tasks to deliver on any real-time demands when they can be interrupted at any moment and for an arbitrary amount of time.
- All Java functions that trigger a rescheduling (such as wait and notify) cannot be called from the interrupt handler. Calling any such function would make the scheduler continue executing a thread or task and you would never be returned to the interrupt handler, which would leave your stack a permanent mess.
- You cannot allow any exceptions to be thrown in the interrupt handler, as this would also result in a messed-up stack. This problem is another reason you would turn off the safety checks. No safety checks means no stack-messing interrupts.

In ADAPT, we turn off the safety checks only for the most performance-critical parts of our drivers and the interrupt handlers.

When writing device drivers, you must have read/write access to memory. Normally, Java does not support direct memory access, but some kind of support for that is obviously needed. The Jbed operating system supplies you with some built-in functions for memory access. To increase the safety of those functions, we encapsulated them once more by creating a set of classes that represents bits, bit-fields, and fields at byte-sized and short-sized memory locations. Listing Eight is one of those classes. It could, for example, be used to represent a value consisting of the bits 4 to 6 (a mask value of 0x70) in a byte-sized hardware register. Using classes like this gives a reasonable level of safety to an otherwise completely unsafe write operation directly to memory. To achieve this and to keep the performance overhead low, as much checking as possible is done in the constructor, which is only called once. When the memory-write function executes, only a simple bounds check is performed. And because we can be sure that the variables in the functions are correct, you can turn off the built-in safety checks for an even smaller performance hit. And with Java exceptions, we can, in many cases, gracefully recover from situations that could otherwise lead to a system crash.

DDJ

(Listings begin on page 86.)

Task	Explanation
com.jbed.tasks.OneshotTimer	Task executed once only.
com.jbed.tasks.PeriodicTimer	Task executed at regular intervals.
com.jbed.tasks.HarmonicEvent	Task executed when another task is started.
com.jbed.tasks.JoinEvent	Task executed when another task has stopped.
com.jbed.tasks.InterruptEvent	Task executed when hardware signals an event.
com.jbed.tasks.UserEvent	Task executed when application signals an event.

Table 1: *Jbed real-time tasks.*

www.diabsds.com

Want some great news?

Introducing DIAB·SDS The New Leader in Embedded Development Tools

Two technology leaders, Diab Data and SDS, have merged and it's a great win for embedded developers! Now you have a single source for the best tools on the market today and the most innovative, open solutions for tomorrow.

Now more than ever, DIAB·SDS is the expert's choice for award-winning C, C++, and Java™ compilers, run-time analysis tools, and powerful SingleStep™ debug solutions. For demanding applications using PowerPC, 68K, ColdFire, M•CORE, Hitachi SH, MIPS, and M32R processors, DIAB·SDS is more than great news - it's great for your design.

diab·sds
An ISI Company
The Embedded Tools Leader

SingleStep™, D-CC™, D-C++™, FastJ™, RTA Suite, SurroundView™

www.diabsds.com

1999, DIAB·SDS, Inc. All Rights Reserved. Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are property of their respective owners.

Listing One

```
public class Task extends java.lang.Thread {
    public static final int timeslice;
    public Task (java.lang.Runnable target, long duration,long allowance,
        long deadline, RealtimeEvent event) throws AdmissionFailure;
 ...
}
```

Listing Two

```
// some real-time constructor is called:
com.jbed.tasks.RealtimeEvent event = new com.jbed.tasks....(...);
// some object implementing the java.lang.Runnable interface is created:
java.lang.Runnable target = ... ;
com.jbed.tasks.Task task;
try {
    task = new com.jbed.tasks.Task(target, duration,
        allowance, deadline, event);
}
catch (com.jbed.tasks.AdmissionFailure e) {
    ...
}
task.start;
```

Listing Three

```
public class LPointer {

    // the two clocks in ms:
    static final int ControllerClock = 5;
    static final int PlannerClock = 10;

    // the calculation parameters:
    static final float Kp0 = 500.0F, Kp1 = 600.0F;
    static final float Kd0 = 35.0F, Kd1 = 10.0F;

    // the Lissajou parameters:
    static float    a0 = 0.05F, a1 = 0.05F;
    static float    f0 = 12.56F, f1 = 6.28F;
    static float    fv0 = 0.0F, fv1 = 0.0F;

    // actual positions and previous positions:
    static float q0actual, q1actual, q0old , q1old;

    // the three tasks:
    static com.jbed.tasks.Task  controllerTask, plannerTask, watchdogTask;

    // the periphery (actuators and counters):
    static com.jbed.peripherals.AnalogActuator out0, out1;
    static com.jbed.peripherals.Counter counter0, counter1;

    // the desired positions:
    static float q0desired, q1desired;

    // the time:
    static float t=0;

    //...

    static void init() {
       //...
       try {
          // Configuration
          float gear = -100F;
          float nmPerAmp = 0.05F;
          float maxAmp = 6.0F;
          float maxNm = maxAmp * nmPerAmp * gear;
          float scale = 2 * maxNm / 4096;

          // define the scale for the actuators:
          com.jbed.drivers.LinearScale.defLinearScale("DAScale",
                                              0, 4095, scale, maxNm);
          drivers.demo.DAC.defModule("DACMod", 0x0FFFF8100);
          drivers.demo.DAC.defChannel("A0", "DACMod", "DAScale", 3);
          drivers.demo.DAC.defChannel("A1", "DACMod", "DAScale", 4);

          // define the scale for the input counters:
          float counterConst = 2000; // = 4 * 500
          scale = 2 * com.jbed.math.FMath.PI / (counterConst * gear);
          com.jbed.drivers.LinearScale.defLinearScale(
              "EncScale", Integer.MIN_VALUE, Integer.MAX_VALUE, scale, 0F);
          drivers.demo.Enc6.defModule("Enc6Mod", 0x0FFFF8300);
          drivers.demo.Enc6.defChannel("Counter0", "Enc6Mod", "EncScale", 3);
          drivers.demo.Enc6.defChannel("Counter1", "Enc6Mod", "EncScale", 4);

          // Initialization
          out0 = new com.jbed.peripherals.AnalogActuator("A0");
          out1 = new com.jbed.peripherals.AnalogActuator("A1");
          counter0 = new com.jbed.peripherals.Counter("Counter0");
          counter1 = new com.jbed.peripherals.Counter("Counter1");
          //...
       } catch (Exception e) {
                    //...
       }
            // ...
    }
   //...
}
```

Listing Four

```
static class Planner extends com.jbed.tasks.Task {
    public Planner(long duration, long allowance, long deadline,
                   jbed.tasks.RealtimeEvent event)
        throws com.jbed.tasks.AdmissionFailure  {
        super (duration, allowance, deadline, event);
    }
    public final void run() {
        q0desired = (a0*com.jbed.math.FMath.sin(f0*t+fv0));
        q1desired = (a1*com.jbed.math.FMath.sin(f1*t+fv1));
        t = t+(float)(PlannerClock/1000.0F);
    }
}
```

Listing Five

```
static class Control extends com.jbed.tasks.Task {
    public Control(long duration, long allowance, long deadline,
                   jbed.tasks.RealtimeEvent event)
                   throws com.jbed.tasks.AdmissionFailure  {
        super (duration, allowance, deadline, event);
    }
    public final void run() {
        float tau0, tau1;
        float dpos0, dpos1;

        q0actual = counter0.read();
        q1actual = counter1.read();
        dpos0 = q0desired - q0actual;
        dpos1 = q1desired - q1actual;

        tau0 = Kp0*dpos0-Kd0*
                ((q0actual-q0old)/(float)(ControllerClock/1000.0F));
        tau1 = Kp1*dpos1-Kd1*
                ((q1actual-q1old)/(float)(ControllerClock/1000.0F));
        out0.write(tau0);  out1.write(tau1);
        q0old = q0actual; q1old = q1actual;
    }
}
```

Listing Six

```
com.jbed.tasks.RealtimeEvent periodicEvent =
    new com.jbed.tasks.PeriodicTimer(ControllerClock*1000);
controllerTask = new Control(2000, 0, ControllerClock*1000, periodicEvent);
controllerTask.start();

com.jbed.tasks.RealtimeEvent harmonicEvent =
    new com.jbed.tasks.HarmonicEvent(controllerTask, 2);
plannerTask = new Planner(1000, 0, 10000, harmonicEvent);
plannerTask.start();
```

Listing Seven

```
001F0F66
          MOVEA.L   (-16, A6), A0
001F0F6A  TST.L     A0
001F0F6C  TRAPEQ
001F0F6E  MOVEA.L   (12, A0), A0
001F0F72  MOVE.L    (-8, A6), D1
001F0F76  ADD.L     (-20, A6), D1
001F0F7A  MOVEA.L   D1, A1
001F0F7C  MOVE.L    (-20, A6), D0
001F0F80  TST.L     A0
001F0F82  TRAPEQ
001F0F84  CMP.L     (12, A0), D0
001F0F88  TRAPCC
001F0F8A  MOVE.B    (-3834, A1), (16, A0, D0.L*1)
001F0F90  ADDQ.L    #1, (-20, A6)
001F0F94  CMPI.L    #$00000008, (-20, A6)
001F0F9C  BLT       $001F0F66
```

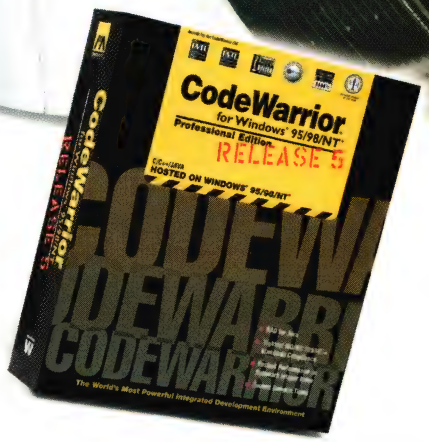
Listing Eight

```
public class ByteRegisterField
{
    private int m_mask;
    private int m_base;
    private int m_pos=0;

    private int m_maxval;
    private static IllegalArgumentException IllegalArgument =
                                        new IllegalArgumentException();
    public ByteRegisterField(int base_adress, int mask)
    {
        m_mask=mask;
        m_base=base_adress;
        //check to see that the mask is in the register and nonzero
        if (m_mask>0xFF||m_mask<=0)
            throw IllegalArgument;
        //Find the lowest bit in the mask
        int i=1;
        while((m_mask&i)==0)
        {
            i<<=1;
            m_pos++;
        }
        m_maxval=m_mask>>>m_pos;
        //check to see if the mask is without holes;
        if(((m_maxval+1)&m_maxval)!=0)
            throw IllegalArgument;
    }
    public void Set(int value)
    {
        //instruct the compiler to turn of safety checks
        com.jbed.runtime.Unsafe.option(com.jbed.runtime.Unsafe.noChecks);
          //check that the value is valid
          if ((value > m_maxval) ||value <0)
              throw IllegalArgument;
          Unsafe.getByte(m_base) & ~m_mask | (value << m_pos)));
    }
    public int Get()
    {
      //instruct the compiler to turn of safety checks
      com.jbed.runtime.Unsafe.option(com.jbed.runtime.Unsafe.noChecks );
      return (com.jbed.runtime.Unsafe.getByte(m_base) & m_mask)>>>m_pos;
    }
}
```

DDJ

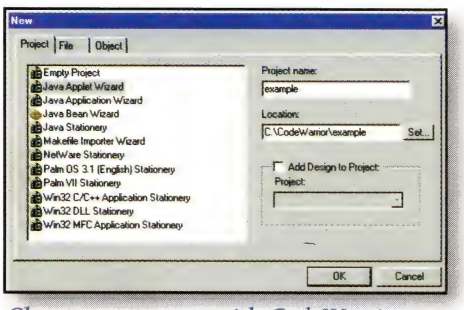
Look good on multiple platforms.



What's cool today may not be cool tomorrow. Make sure you're ready when the platforms change (and we're not talking shoes, here).

Stay on your feet, no matter what they throw at you.

CodeWarrior for Windows is the only Windows-hosted development toolset that lets you target multiple platforms. Code for Windows today, Mac tomorrow, perhaps Palm next week. How about NetWare, game consoles or embedded systems? Yep, you can do it all with CodeWarrior. Working on a different platform at home and work? CodeWarrior lets you port your projects between host platforms without missing a beat.



Choose your target with CodeWarrior.

Put on your running shoes.

The best part is how fast you can get to work. No long learning curves just to hit a new target.

Look at these CodeWarrior benefits:

- ✔ The state-of-the-art IDE helps you *shorten your development time*.
- ✔ CodeWarrior allows you to *port projects* between platforms.
- ✔ *Everything you need...* project manager, editor, compilers, debuggers, linkers, assemblers, a file compare utility, and the flexibility to plug-in third party components.
- ✔ Tightest ISO/IEC *C++ standards compliance*.

See for yourself.

Get your FREE* 30-day evaluation copy and put it to the test. Plus, you'll receive **7 books on programming, a $200 value, included FREE on the CD.** You've got nothing to lose and everything to gain!

CodeWarrior vs. The Rest

Platforms	CodeWarrior	Visual C++	C++ Builder
Windows 95/98/NT	✔	✔	✔
Mac OS	✔		
Linux	✔		
Solaris	✔		
Palm	✔		
NetWare	✔		
Nintendo 64	✔		
Dreamcast	✔		
PlayStation	✔		

*CodeWarrior supports many more platforms and targets.***

FREE!* 7 Books $200 value

Discover why over 200,000 programmers worldwide use CodeWarrior.

Call today or order online
FREE 30–day evaluation

http://www.metrowerks.com/go/ddj

1-800-377-5416

*Shipping and handling of $1.95 per CD applies, plus state sales tax if applicable.

Hurry! This offer expires 12/31/99.

metrowerks®

CodeWarrior® for Windows

© 1999 Metrowerks. All rights reserved. Metrowerks, the Metrowerks logo and CodeWarrior are registered trademarks of Metrowerks Inc. Visual C++ is a product from Microsoft Corporation, and C++ Builder is a product from Borland/Inprise. All other products are trademarks of their respective companies. All prices are in US dollars; offer good in US and Canada only. **CodeWarrior for Windows 95/98/NT, Professional Edition, targets Windows 95/98/NT and Mac OS only; other targets available separately.

Machine Learning & Agent-Based Computing

AI finds a home on the Internet

Zhimin Ding and Li Liu

As the Internet becomes ubiquitous, software is becoming increasingly exposed to complex and dynamic environments. Consequently, the concept of agent-based computing has emerged as a useful model for software that has to deal with an uncertain and changing environment that consists of many distributed software systems.

Agents are software entities that not only gather and present information, but also act intelligently to accomplish one or more assigned goals. Agents can often be described as finite state machines whose state transition dynamics drives them to achieve the goal.

A state-machine model whose state transition dynamics has to be entirely hardwired and manually maintained offers little value to programmers. In this article, We'll examine the application of machine-learning technology to control an agent. Based on Markov process, this technology enables an agent to automatically identify the statistical structure of its environment and synthesize optimal feedback strategies at run time. As the result, a software agent powered by this technology will be able to adapt to its environment and continuously learn and improve its ability to achieve the goal. To this end, we present MLEngine, a general-purpose AI engine with real-time learning capability. MLEngine can be used to control simulated game characters or software agents through interface with (simulated) sensors and actuators. MLEngine can synthesize goal-oriented behavior at run time and therefore make the characters under its control learn. MLEngine, available from both *DDJ* (see "Resource Center," page 5) and at http://www.lmtg.com/, includes Win32 run-time libraries, API documentation, sample source code, and demo programs.

Agent Dynamics

Again, the dynamics of an agent can often be conveniently described as a state machine. The basic elements of a state machine are:

- At any instance, the dynamic state of an agent can be uniquely described by a set of variables called "state variables." These variables can be analog or discrete.
- An agent can receive input stimuli from its environment, and the state variables will undergo changes. The new state is a function of both the environment input and the previous state.
- An agent can pass stimuli back to its environment. The output is uniquely determined by the state variables. The environment could include other agents.

If the states are discrete and finite, this mechanism is called a "finite state machine."

An intuitive way to illustrate the idea is to use a game character such as that in Figure 1. Suppose that as this game character moves around in space, you could use its position in space to describe its dynamic state. Additionally, a character usually has an orientation. An orientation indicator can serve as another state variable. Imagine that this character can also sense whether it is in touch with an obstacle. Touch-sensing input can also become a state variable. So to completely describe the character, you need to specify its position, orientation, and whether or not it is in touch with an object.

The state variables undergo changes over time. For example, the character can move in space and flip. The way it moves can be expressed as transitional probability between states. Obviously, the agent state transition will be influenced by inputs from its environment under certain conditions, so that it will not become a completely self-indulging entity. For example, the character's position change depends on its environment, such as whether or not an obstacle is present.

The character's state should also be able to undergo changes without any environment stimuli under certain conditions (so that the new state depends only on its previous state). This is important for implementing any oscillatory or self-driven sequencing behavior such as walking in a legged character.

Assume that you are interested in an agent whose behavior could evolve as it adapts to its environment. A sensible approach toward that goal is to separate constraints from voluntary control in the agent design. A suitable analogy is the human model. Human behavior is determined partly by the brain and partly by the constraints imposed by the physical body (for example, you couldn't jump 100 feet in the air even if you wanted to).

In Figure 2, you divide an agent into a physics block and a control block. The

Zhimin is the founder of the Learning Machines Technology Group and can be contacted at learning@lmtg.com. Li is a software design engineer at Hewlett-Packard. She can be reached at liliu@hpl.hp.com.

Wishing for a directory that supports all of these?

[If so, you're not alone.]

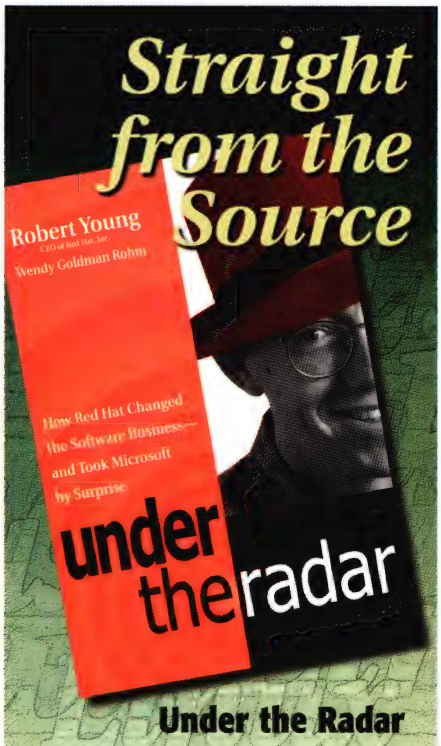
That's why many of the world's leading companies have turned to Novell Directory Services® (NDS™). When you incorporate NDS into your applications, some pretty amazing things happen to your products. They get to market faster. They communicate more easily with other applications. And they become secure over the Internet.

Just ask companies like Lucent Technologies* and Nortel Networks*. They'll gladly tell you how implementing NDS can add a whole new dimension to your applications. Because they'll be able to draw upon a common set of user profiles and data on the network. Which simplifies user access, tightens security, and reduces the time and cost of administration. All of which make your programs more effective, whether you're developing for custom or commercial applications.

Directories are the future, and NDS is the present and future of directories. To see why you should be a part of this movement, visit http://developer.novell.com/nds/ today for online demos, software development kits, a DeveloperNet subscription and more.

Novell®

99 Novell, Inc. All rights reserved. Novell, Novell Directory Services, and DeveloperNet are
tered trademarks and NDS is a trademark of Novell, Inc., in the United States and other countries.
ent Technologies is a registered trademark of Lucent Technologies, Inc. Nortel Networks is a
mark of Northern Telecom Limited.



Under the Radar

How Red Hat Changed the Software Business—and Took Microsoft by Surprise

By Robert Young, CEO of Red Hat, Inc. and Wendy Goldman Rohm

ISBN: 1-57610-506-7 • $27.50 U.S. • $37.50 Canada

ON SALE NOW

Under the Radar is an adventure story about the surprising rise of Red Hat, Inc. and the Open Source Movement.

What the reviewers are saying...

"...required reading for anyone who thinks that the software wars are over."

—REBECCA LYNN EISENBERG
CBS.MarketWatch

"Wendy Goldman Rohm and Bob Young have written a thriller. Their fast-paced account of how Linux is subverting the trillion dollar software industry reads more like Grisham than computer code. It will send shivers down readers' necks, especially if they are called Bill Gates."

—DANIEL JEFFREYS, London Daily Mail

CORIOLIS™

Available at Book and Computer Stores Nationwide
Telephone: 800.410.0192 • Fax: 480.483.0193
International callers: 480.483.0192
In Canada: 800.387.9776

www.coriolis.com

©1999 The Coriolis Group, LLC. All Rights Reserved. DD1199 999
Prices and availability subject to change.

(continued from page 88)

physics part of the agent is a state machine. The controller has access to state information and issues a set of actions. Along with the other inputs, the action input will influence state transitions. So now the state transition is determined not only by external input and the previous state, but also the action input from the controller.

The character in Figure 1, for example, has five movement actions—left, right, up, down, and flip. The probability of the character falling into any state depends not only on its previous state, but also on the environment input as well as the voluntary action input from the controller.

Setting Up the Goal

Before learning can take place, it is necessary to come up with a formal definition of a goal. After all, the purpose of learning is for the agent to achieve the goal.

A convenient way to define a goal is to choose one or more states and assign them as the goal. The fact that the goal is expressed as one or more states means that the goal has to be a part of the state space that can be sensed by the controller. Intuitively, this makes sense because no matter how intelligent an agent is, it would never be able to learn and improve if it cannot sense whether or not it has achieved the goal.

To make learning happen, you have a few obvious and tempting choices:

- You simply observe and see which actions cause the state transitions that lead to the goal. You then enhance the probability of taking these actions. This is essentially a conditioning based on reward and penalty. You reward the actions that bring favorable results and penalize those that do not.

  The problem with this approach is that learned behaviors tend to be too simplistic. The character can only learn simple reflexes that seek immediate rewards. It will not learn to plan ahead and carry out sequential behaviors.
- You try different combinations of features in the feedback strategy and see which one yields the best overall result. If we try long enough, a good combination of features in the controller can be found. And the character could become very smart.

  Experience suggests that combinatorial approaches only work well for simple problems. As the state space grows big, the amount of time needed to perform trial-and-error quickly becomes intractable, even if it is done in a simulation environment.

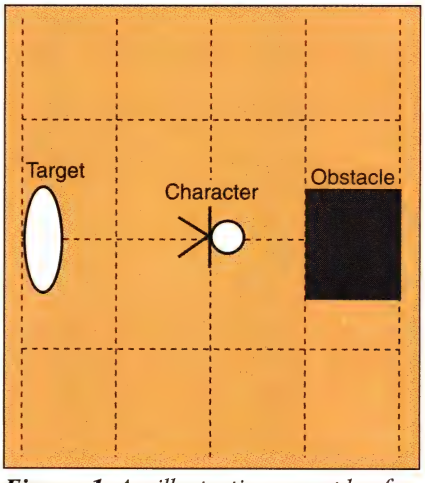


Figure 1: *An illustrative example of an agent.*

Mathematical Analysis of Learning

For convenient analysis, you can encode the states in the state space into a linear array 1,2,3...n, assuming there are a total of n mutually exclusive states.

Once the states are encoded into a linear array, the state transition can be described by an $n \times n$ matrix. Since the state transitional probability depends on the action input from the controller, to completely describe the physics of a character you need m such $n \times n$ matrices, assuming there are m actions. In Example 1, for instance,

$$a_{ij}^{(k)}$$

represents the transitional probabilities from state i and state j under action k. If the state transition is fully deterministic, then a_{ij} is either 1 or 0; otherwise, it is some positive value in between.

In Example 2, the controller can be described by an $n \times m$ matrix, where x_{ij} is the probability of taking action j when the character is in state i.

The combined system that includes both the physics and the controller can be described by a transitional probability matrix T (see Example 3), where

$$t_{ij} = \sum_{k=1}^{M} x_{ij} \bullet a_{ij}^{(k)}$$

Suppose that on average, the state probability can be described by $P=\{p_1, p_2, \ldots p_n\}$, where p_i is the longterm average probability for the system to be in state i.

According to Markov process theory, under a certain condition (the ergotic condition), the state probability will converge to a stable value that satisfies $P=P \bullet T$.

Suppose that the goal is state o, the purpose of learning is to enhance P_o by manipulating control variables. Obviously, you cannot directly manipulate P, which is the consequence rather than the cause of the system behavior. You can, however, manipulate transition probabilities t_{ij} through x_{ik}. This will, in turn, influence P in the long term, based on the relationship $P=P \bullet T$.

The Power of UNIX on WINDOWS!

NOW ALL YOUR UNIX KNOWLEDGE IS JUST AS USEFUL ON WINDOWS ...AND SO IS YOUR UNIX CODE!

Are you a UNIX® pro working on Windows®?
Developing? Deploying? Administering?

NOW ON WINDOWS YOU CAN:

- Use a UNIX command-line environment: shells, commands, and utilities
- Run UNIX scripts
- Port business-critical UNIX programs
- Do cross-platform development—write once & deploy on both Windows and UNIX
- Web-enable existing applications
- Interoperate with UNIX using X, NFS and telnet
- Integrate mission-critical and workgroup applications onto a common desktop

Choose MKS Toolkit®, Toolkit Select, or N*u*TCRACKER® Professional. All these products have been proven to cut learning, development and support time—70%, 80%, even 90%. And there are thousands of customers and case studies to prove it.

TO ORDER YOUR 30-DAY RISK-FREE TRIAL OFFER, CALL 800-637-8034 OR 703-803-3343 OR VISIT US ON THE WEB AT WWW.MKS.COM/INTEROP

ORDER YOUR FREE PORTING GUIDE TODAY!

NuTCRACKER®

mks®

MKS, MKS logo, MKS Toolkit, NuTCRACKER and NuTCRACKER logo are registered trademarks of Mortice Kern Systems Inc. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Limited. Windows, Win32, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries.

End the Chaos...

SOFTWARE ENGINEERING

@Carnegie Mellon

The Discipline of Software Engineering.

Master of Software Engineering in Pittsburgh, New York City, and Washington, D.C.

Certificate in Software Engineering, graduate or continuing education courses by distance delivery.

Visit our website.

http:www.distance.cmu.edu/moreinfo/

Everyone's reading Linux Journal.

Find out why.
[for free!]

Request a free sample issue @ www.linuxjournal.com
[Free issue will be shipped with our next available mailing, or added to the end of your paid subscription.]

(continued from page 90)

In other words, the goal of optimization is to select the appropriate x_{ik} so that p_o is at its maximum.

There is one missing piece in this analysis. The learning algorithm depends on knowledge about state transition probabilities

$$a_{ij}^{(k)}$$

So, before learning could occur, you need to fill in the state transition matrices.

As mentioned earlier, state transition probability depends on the character's physics, as well as its interaction with its environment. The process of trying to find out state transition information from knowledge about the character's physics is called "system identification."

Our approach here is to come up with a run-time algorithm that calculates state transitional probabilities by observing the character's physics in action.

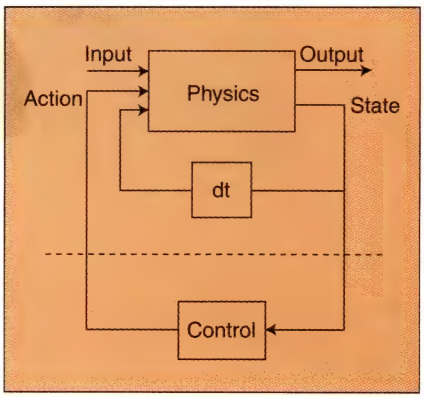


Figure 2: *The relationship between physics and control. State transition is determined by external input; previous state; and action input from controller.*

$$A^{(k)} = \begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & \cdots \\ a_{21}^{(k)} & a_{22}^{(k)} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & a_{nn}^{(k)} \end{bmatrix}$$

$k=1,\ldots,m$

Example 1: *Transitional probabilities.*

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & \cdots \\ x_{21} & x_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & x_{nm} \end{bmatrix}$$

Example 2: *Describing the controller by an* n×m *matrix.*

code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.
code.code.code.code.code.code.code.code.
boring.boring.boring.boring.boring.boring.

Announcing the end of boring code.

The all new digital media operating system from Be.

It's here...www.be.com/drdobbs

(continued from page 92)

The advantage of a run-time approach is that the controller is adaptive. If the physics of the character and its environment undergo changes, the controller will reorganize and adapt the changes. The end result is that the character can maintain and improve its capability to achieve its goals in response to environment changes.

Here is how system identification works. The controller issues actions and observes changes in state. If action k is issued and the state undergoes a change from i to j, you will make adjustments to

$$a_{ij}^{(k)}$$

in Example 4, where λ is the learning rate. It is usually a small positive number less than 1. For all the other

$$N-1\ a_{i'j}^{(k)}$$

with $i' \neq i$, you make adjustments according to the amount in Example 5. If you keep doing that, over a period of time, matrices

$$\{a_{ij}^{(k)}\}$$

converge to best represent the state transition probabilities.

Putting It Together

Based on this analysis, you have put together an agent controller such as MLE, which can be integrated into an agent and make it capable of learning. The learning is carried out in stages.

In the explore phase, MLE drives the agent to interact freely with its environment. MLE passes out quasirandom action commands and monitors the state input. As a result, the agent will start to explore its environment. Internally, MLE associates sensory and action information and builds an internal representation of the agent and its environment.

$$T = \begin{bmatrix} t_{11} & t_{12} & \cdots & \cdots \\ t_{21} & t_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & t_{nn} \end{bmatrix}$$

Example 3: *The combined system.*

$$\Delta a_{ij}^{(k)} = \lambda \bullet (1 - a_{ij}^{(k)})$$

Example 4: *Making adjustments.*

$$\Delta a_{i'j}^{(k)} = \lambda \bullet (0 - a_{i'j}^{(k)})$$

Example 5: *Making further adjustments.*

This is essentially a nonmodel-based system identification process.

A parameter called "association entropy" can be obtained from MLE by calling *GetAssociationEntropy*. This value (which ranges from 0 to 1) should go down as the association phase is being carried out. The smaller this value is, the more deterministic MLE has become while it builds an internal representation of the system under its control. It represents the confidence level of the agent in terms of its understanding of the environment.

In the optimization phase, MLE will synthesize a strategy, which drives the agent to achieve the goal defined by *SetGoal*. The strategy is essentially a sensory feedback controller that determines how the system should react to sensory inputs and interact with its environment.

A parameter called "optimization entropy" can be returned from the MLE by calling *GetOptimizationEntropy*. This value should go down as the optimization phase is being carried out. The smaller this value is, the more deterministic the synthesized strategy is. The estimated probability to achieve the goal can be obtained from *GetGoalProbability*.

The three phases can be switched back and forth by monitoring association entropy, optimization entropy, and the goal probability. For example, during the optimization phase, the application can keep statistics of the average probability of achieving the goal and compare it with the estimated goal probability returned by *GetGoalProbability*. If the two are sufficiently different, the application can start to explore and associate again, during which the association entropy is monitored. Once the association entropy goes down to a stable value, an optimization phase can be started. The optimization entropy should go down while the goal probability will go up; see Figure 3.

We've included two demos (available electronically) of agent learning. The first is a boxer trying to hit its opponent. At first, the opponent does not really respond well to the boxer's moves. The boxer learns to carry out coordinated arm movement to hit the opponent repeatedly at the highest frequency. Later, we change the opponent's strategy by making it more responsive. The opponent moves his fist to block the boxer's punch. After some adjustment, the boxer learns to fake fist movements and trick the opponent to move its fist to the wrong location, and then hits from another direction.

The second demo is Picker, a cartoon character whose goal is to eat the fruit at various locations. It learns to associate its visual information with its movement and approach the fruit efficiently. You could cause Picker to mutate so that its mouth and its eye will be located on a different side of the body. The picker learns to flip around so that it can see where the fruit is and approach it with its eye facing the fruit. Once there, it turns around again and eats it. For a Pentium-class PC, it usually takes one or so minutes to learn a new strategy if the size of the state space is less than 50.

We have also tried MLE with (analog) simulations that faithfully emulate physics. For example, we coupled MLE with a quadruped simulator that simulates human body mechanics. we set the goal as the model making forward motion while maintaining an upright body position. The model learned to walk and run realistically like a human (see Figure 4).

The MLE API

Based on the aforementioned method, We've designed a general-purpose AI engine called MLEngine (MLE); its interface functions are presented in Listing One. To prepare the character for learning, you need to follow these steps:

1. Design the character's physics.
2. Encode the state information into a

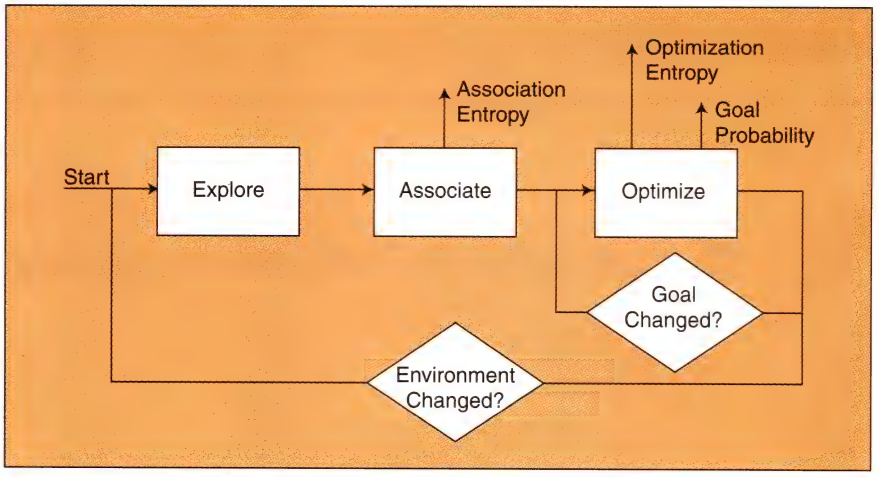


Figure 3: *MLE flow chart.*

YOU'LL NOTICE SOMETHING DIFFERENT WHEN YOU DOWNLOAD OUR XML INDEXING ENGINE.

STARTING WITH THE FACT THAT YOU CAN ACTUALLY DOWNLOAD AN XML INDEXING ENGINE.

Affordable XML is available now.
Download Xdex™ for $495 and be up and running in just 30 minutes.

www.xmlindex.com/dd

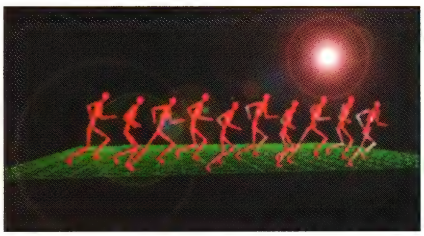


Figure 4: *Agent controller coupled with real physics to model human locomotion.*

(continued from page 94)

linear array ranging from 0 to *dim_state -1*. For example, if an agent has three state variables, each represents three possibilities. It is sufficient to represent the dynamics of the agent into 27 discrete, mutually exclusive states (or less since some combinations may not be possible).

3. Create an instance of MLEngine. Specify the dimension of the state space and the action space.
4. Select one or more states as the goal and call *SetGoal* to set them up. In the boxer example, the goal is the state when the arm is extended and the fist is landed on the opponent's body.
5. Connect the controller to the character physics by calling *Perform*, which accepts state information as the input and pass out actions.

The learning process is carried out in three phases (see Figure 3). Phase control is accomplished by calling *SetPhase*.

Comparison with Other Approaches

At the beginning of AI, intelligence was thought of as the ability to manipulate symbols and abstract concepts. An intelligent system based on this assumption consists of a reasoning engine and database that stores common sense knowledge (an expert system, for example). To make it useful, real-world information has to be mapped into abstract concepts or symbols before machine reasoning can take place and the expert system database can be updated. The task of recognizing real-world patterns and mapping them into concepts often turns out to be prohibitively difficult.

Symbol-based automatic reasoning is becoming less popular these days because it never produced interesting results. Recently, a school of thought has emerged that believes logical reasoning is more an afterthought that justifies actions that are already taken, than the cause of actions. For example, when you face a situation and make a decision, it is often after you have made that decision that you come up with the thought: "I did that because of such and such reasons..." We all have a tendency to consciously justify what has been decided unconsciously.

The literature on neural networks is vast. The most popular models focus on pattern association and classification. For example, back-propagation models can map a set of input patterns into some output patterns according to a set of examples. A back-propagation neural net can be used as a controller that maps sensory input patterns into actuator outputs provided one supplies examples about what to do under all situations. The controller can only generalize from these examples.

A second class of neural net models can learn to categorize analog patterns into discrete clusters without supervision. The purpose of learning is to reduce encoding error. The best known example is the Kohonen neural net model.

There is yet another class of neural net models that are able to generate sequences of spontaneous patterns. The output of these neural net models can be used to control motor actions. Most of these neural net models have recurrent topology and dynamics (Hopfield neural networks and Bolzman machine, for instance). Markov process theory has often been used to mathematically describe the dynamics of recurrent structures. Our learning algorithm is similar to some aspects of these models, but only at the mathematical level.

Learning automata are a set of mathematical models that address learning directly without worrying about the underlying brain substrate. The heart of this approach is conditioning based on reward-and-penalty. The main problem with this class of methods is that the results are usually too simplistic. For example, a typical learning automaton can only learn to perform actions that will directly lead to reward. It will not learn to plan ahead and accomplish the goal through indirect action sequences.

More sophisticated learning automata are being developed today. Most of them try to give credit to actions that indirectly contribute to achieving the goal. Some of these models borrow ideas from neural network models.

It is said that if you give a monkey a typewriter and allow it to type randomly, you will eventually see beautiful poems coming out of that typewriter. Many tough problems can indeed be solved by random searches in the solution space if we do not worry about the time it takes. Simulated annealing and genetic algorithms (GA) are two such techniques. Combinatorial optimization techniques such as GA and simulated annealing search through the control parameter space to optimize the performance measured by a predefined index or criterion. The search is usually guided by some sort of heuristic. For example, simulated annealing uses gradient information to guide the search. GA recombine simulated gene segments to come up with control parameters for new trials.

The main problem with this class of approaches is the computational requirement. For example, if you use GA to optimize the performance of a game character, the algorithm has to simulate the game character through many generations. In each generation, a large population of individual characters has to be simulated also. The algorithm will come up with many sets of control parameters for those individual characters. Each of these characters has to perform long enough to come up with a fitness measurement. A character will not adapt in its lifetime. Combinatorial algorithms are fundamentally difficult to implement in real time.

Conclusion

There is not a one-size-fits-all solution for machine learning. A practical solution has to inherit the best of what is available. Learning as an optimization algorithm may take many forms. A successful one converges to an optimal result not because it is a neural net, learning automata, or whatever else, but because its underlying mathematics is correct. It is therefore best to deal with learning at the mathematics level.

DDJ

Listing One

```
typedef state int;
typefef action int;

class MLEngine {
public:
  MLEngine(int dim_sate, int dim_action, int dim_order);
  ~MLEngine();

  void SetGoal(state);       //one of the states is assigned as the goal
  void SetGoal(state*, int); //when multiple states are assigned as the goal
  action Perform(state);     //0--explore, 1--associate, 2--optimize and perform
  void SetPhase(int);

  void Associate();
  double GetAssociateEntropy();

  void Optimize();
  double GetOptimizeEntropy();
  double GetGoalProbability();

  void SetAssociationRate(double r);      //0<r<1
  void SetOptimizationRate(double r);     //0<r<1
}
```

DDJ

Where will your best ideas end up?
That's the real question, isn't it? It's not whether you have the ideas; it's what's going to happen to them. The truth is, great ideas are generated all around the world every day. Some land in the trash can, some get trampled in meetings, and most of them get put on the back burner. Indefinitely. Then there are the ideas that survive. The ones that end up doing things, accelerating technology, and even changing lives. So what makes the difference? Being part of an environment that is able to make ideas happen. You've got to be with a company that has the resources, the talent, and most important, the worldwide influence to turn your ideas into reality. Every incredible innovation starts as an idea. So does every pipe dream. See where your ideas could go, at:
www.ibm.com/whywork
IBM
IBM is committed to creating a diverse environment and is proud to be an equal opportunity employer. IBM is a trademark of International Business Machines Corporation. © 1999 IBM Corp.

Trends in High-Tech Hiring Practices

Finding your way in a rough and tumble job market

Reginald B. Charney

Hiring practices in the information technology (IT) industry are changing more rapidly than ever before. To make sense of this rapidly changing employment landscape, I compiled—then analyzed—a database of over 100,000 new technical jobs listed on the Internet. In 90 days, from the beginning of May 1999 to the end of July 1999, an average of 219 new technical positions were listed daily. Of this number, 114 were for positions as software engineers, which accounts for more than 52 percent of all new IT jobs listed. In that period, the number of technical positions advertised rose 17.4 percent, while software engineering openings rose 49 percent.

In this article, I present a number of charts that illustrate some trends my analysis reveals. While the charts focus on technical skills that are in demand, opportunities also exist in areas such as management and marketing. Consequently, I'll touch on those areas, as well as share some pointers on maneuvering in today's job market.

Reg is president of stats2go.com. He is also a voting member of ANSI's X3J16 Committee on C++. He can be reached at reg@stats2go.com.

Technical Skills Are in Demand

The charts I present show the demand growth rates for skills—not the total number of positions available. For example, if there was a demand for six Eiffel software engineers at the start of May and a demand for 12 at the end of July, then demand for Eiffel skills increased 100 percent. On the other hand, if during the same period the demand for Java software engineers increased from 5000 to 6000, then the growth rate is only 20 percent. Negative growth rates indicate shrinkage in demand for that skill.

In developing these charts, I've made a number of assumptions. First, the trend lines have been treated as linear. Also, this sample is representative of the job market in general. Lastly, we know that hot topics like Linux and Java have nonlinear growth rates, but for this short a time period, a linear assumption provides a good estimate.

Figure 1(a), for instance, shows that jobs for PL/1, Cobol, and SmallTalk programmers are losing ground as demand for them dwindles. On the other hand, demand is growing fastest for XML/HTML, Eiffel, Perl, and Java developers.

Figure 1(b) illustrates the normalized demand for each language. Here, the market growth/shrinkage is multiplied by the proportion of the market the language has. At the moment, more opportunities abound in Java and C/C++ than in any other language.

Figure 2(a) shows that the growth rate for Linux skills is outstripping demand for any other platform. Meanwhile, demand is up for skills on all major platforms. (Both UNIX and Macintosh trends are based on regression calculations, not on actual figures.)

In Figure 2(b), it appears as if the current demand for Windows NT, and Windows in general, far outweighs any other platform. However, Solaris and Linux both make a respectable showing. Especially since the latter has only recently received a lot of media attention. (Both UNIX and Macintosh normalized demands are based on regression calculations, not on actual figures.)

Figure 3(a) illustrates the changing demand in different technologies. The growth rate is greatest for ASIC-, database, CORBA-, and Internet-related skill sets. Internet-related skills are defined as positions listing Internet, Java, HTML, DHML, or XML in their job listings.

Finally, Figure 3(b) shows that the greatest demand is in Internet-related technologies. Database skills come in a respectable second. While the first is no surprise, the fact that database skills are in high demand may be.

On the Management Front

It is no secret that, in all business areas, outsourcing is becoming a way of life.

THERE IS A PERSON INSIDE ME

WHO WANTS

to invent something

to help the homeless

to pay off my loans

to meet Colin Powell

to fly in space

to write a book

to use my degree

to try base jumping

There are plenty of graduates with good grades, but we're looking for students with dreams and ambition. People who want to take on challenges; build jetliners, fighter planes, the Space Station. People who want to be an integral part of the projects they work on. If you're studying engineering, computing, or another technical discipline and have big dreams to go along with your degree, submit your resume to Boeing at www.boeing.com/employment or mail it to The Boeing Company, P.O. Box 3707, M/C 6H-PR, 003705, Seattle, WA 98124-2207.

It is the policy of The Boeing Company to attract and retain the best qualified people available without regard to race, color, religion, national origin, gender, sexual orientation, age, disability, or status as a disabled or Vietnam-era veteran

www.boeing.com/employment

BOEING

(continued from page S2)

Companies are doing this so that they can concentrate on their "core competencies" and become more efficient. All other functions are being farmed out to contracting organizations. This includes human resources, payroll, system maintenance, operations, and most low-tech jobs. Labor and tax laws also make contracting as an individual activity more and more difficult. Firms do not want the potential liability of hiring independent contractors, so they buffer themselves by using outsourcing agencies as much as possible. In many high-tech firms, contract staff compose 14 to 40 percent of the work force. Because many companies are hesitant to reveal exact outsourcing figures, these estimates are probably low. Also, the outsourcing trend will grow. In the future, expect it to exceed 70 percent of the operating staff at many companies. Only the timeframe is in doubt.

Outsourcing also tends to turn contract staff into commodities. Because companies are using outsourcing for noncritical tasks, the outsourcing agencies are looking for skilled workers who give reliable and consistent results. To avoid employment liability, companies also may only keep contract staff for a short period and expect the outsourcing firm to supply replacements on a regular basis. This employment by the numbers also destroys any concept of company loyalty and the idea of commitment to a goal. More and more contractors will treat the job as a 9-to-5 experience.

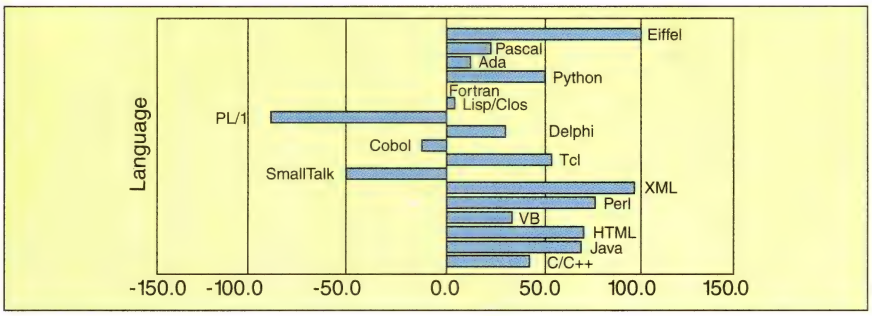


Figure 1(a): *Three month demand trend in languages.*

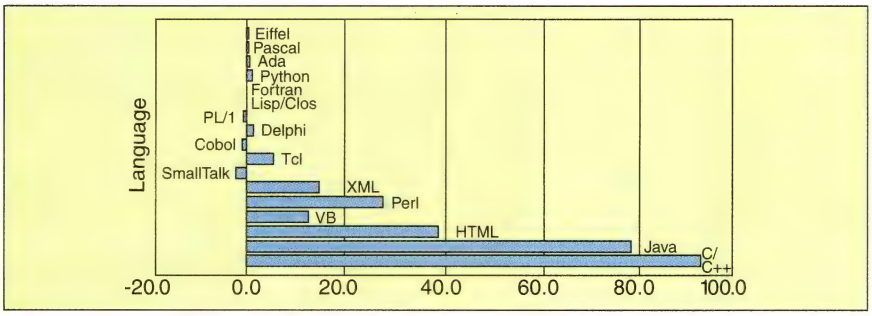


Figure 1(b): *Normalized demand by languages.*

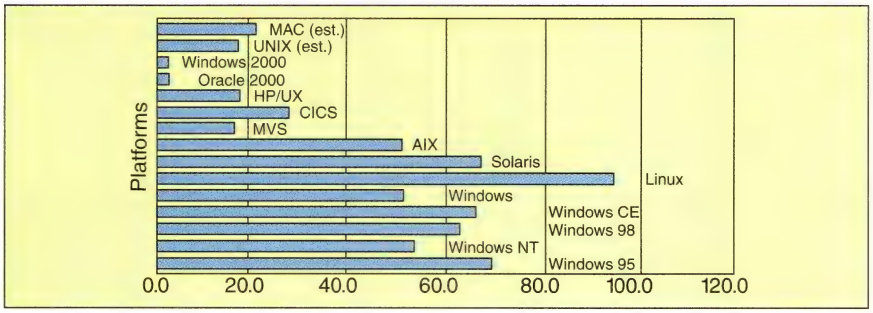


Figure 2(a): *Three month demand trend in platforms.*

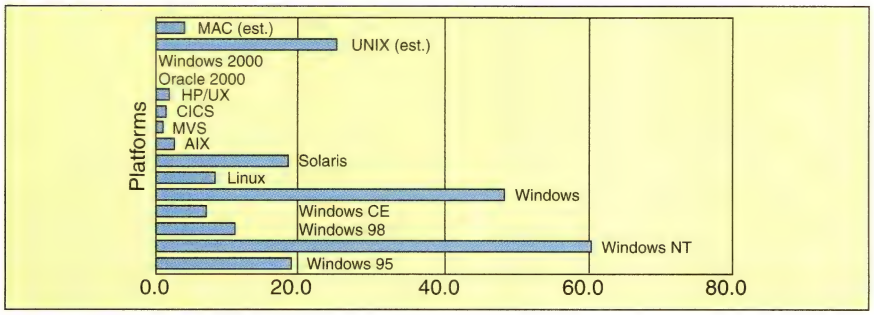


Figure 2(b): *Normalized demand by platforms.*

Outsourcing is not necessarily bad for individual contractors. People working for outsourcing companies get more varied experience and higher basic compensation than their directly employed counterparts. Contractors also get their marketing done for them by outsourcing firms, and contractors do not need to spend significant amounts of time finding the next contract nor dealing with the complex sales aspect of selling their services.

In Marketing

Marketing makes the world aware of your products and the potential solutions that your products and services offer the customer. To do that, marketing tries to find out what customers want. There are many ways of doing this, but the growth of the Internet has added new dimensions to this effort. It is now possible to get tens of thousands of potential customers interested, where before marketing was limited to dealing with hundreds at most. It used to take weeks, even months, to find out what was happening. It is now possible to get useful information in hours or in a day or two. Given that marketing has found out what is needed, they then want to sell it on the Web using e-commerce. This will affect the basic technology we use and how we use it. Massive system rewrites will occur—not in the sense of modifying code, but in creating new systems to replace old ones. The new systems will also be expected to have greater functionality, interoperability, flexibility, efficiency, and reliability.

Marketing may be the first to see the business opportunity for these new systems, but expect the company to increase the pressure on its IT staff so that the company can stay competitive. To deal with the explosive growth that the Internet brings, best-of-breed software will be purchased and deployed. So, besides needing to garner and improve your overall programming skills, expect to be trained in new externally developed products that your company will purchase/lease.

Hiring Efficiency

Job seekers are using the Internet more and middlemen less. At the same time, those hiring are becoming inundated with information and résumés. Both parties are also going nationwide using the Internet. Decisions on interviewing and hiring are

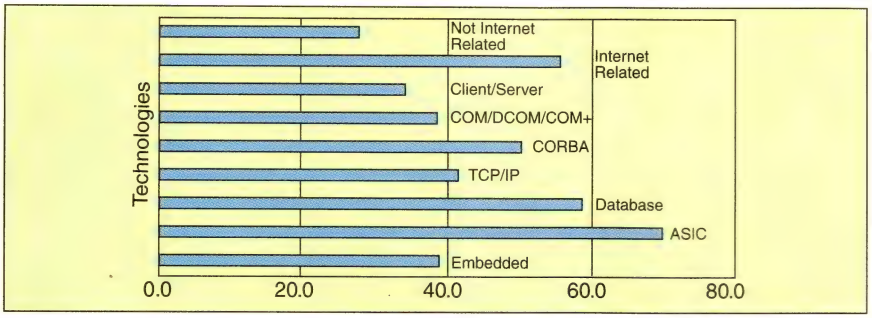


Figure 3(a): *Three month demand trend in technologies.*

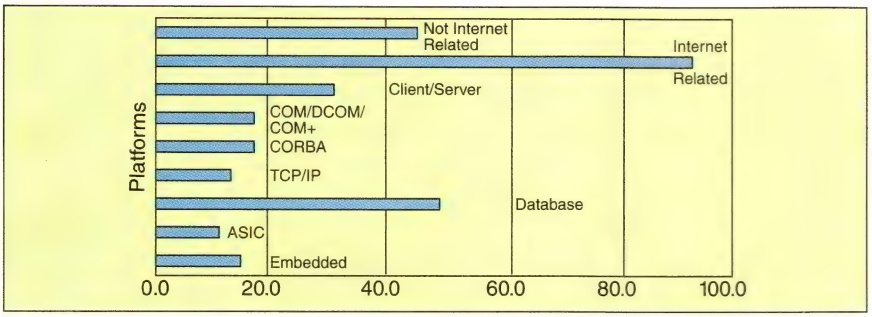


Figure 3(b): *Normalized demand by technologies.*

speeding up — in more and more cases, you will hear in 1–2 days if you are in the pool of eligible candidates or not. The first interview is also likely to be a telephone interview. These trends are happening in many high-tech industries, not just computing. As Bruce Hatz, Hewlett-Packard Corporate Staffing Manager said, "Rapid response to qualified applicants is a competitive advantage." Thus, the question becomes "Are you a qualified candidate?"

The ideal candidate is a self-motivated, self-starting, team player who has just done exactly the desired job successfully in a closely related business — preferably at the competition (if those hiring you can avoid a lawsuit). Such a candidate is ideal because they are the ultimate in efficiency: no training, immediate utilization, highly recommended (by the competition), and no wasted effort on the part of the hiring manager trying to balance various tradeoffs. They are worth even more than they were getting before because they shorten and simplify the hiring cycle, degrade the competition's competitive advantage, and add to the hiring manager's reputation for attracting new people.

Unfortunately, there never was nor ever will be an ideal candidate as described earlier. Any new employee is initially inefficient: The new company's culture differs from the old one; the new team takes time to integrate any new team member; and the new firm's processes and procedures differ from the old firm's way of doing things. It will take any new employee three to six months to integrate and become a part of any effective working group. Also, consider this: If the new person has done the same job at his previous employer, then he may not bring any new approaches. Remember that this person was hired because he didn't rock the boat — he was supposed to be plug-n-play.

We are constantly faced with the tradeoff between short and long term goals, between geeks up on the latest and greatest technology and those with a broader vision gained through experience. There is a place for both, but there are more slots for geeks than there are for the person with the vision that experience brings. You can deduce this from the fact that there is only one CEO at a company, but many individual contributors. So where does someone with broad experience go? My recommendation is into middle management, project management, and mentoring. Or as one CEO of a rapidly expanding startup said, the older managers provide adult supervision for the younger employees.

Middle management is currently a dirty word, but newer high-tech companies now face a dilemma. They have few management levels and the manager at each level tries to manage more and more people. Industrial psychology and organizational dynamics studies show that the ideal small group size is 7 +/- 2 people (Miller's Law). Graicunas' Law states that relationships increase exponentially with the number of people managed. Thus, a department larger than 150-200 people becomes unmanageable. A department of 200 needs to be broken up into 20–30 teams. Each team uses a team or project leader. A possible management hierarchy could be one department head, 4–5 assistant department heads, 25–30 project leaders, and 164–170 individual contributors.

Prepare Yourself

Here are a number of things you can do to actively manage your career:

- Learn at least one new skill per year and use it so that the experience is real.
- Network inside and outside your normal environment. Volunteer for something that interests you.
- Be opportunistic. Learn about what is hot and what is not.
- Find a mentor, be a mentor.
- If you are transitioning up the organization, keep your skills set current for the type of team you are managing — don't give up your day job.
- Keep your résumé as current as possible. List skill sets, responsibilities, and projects as you work on them.
- Be prepared to contract with an outsourcing agency — this means keeping your skills current.

Here are some things to avoid:

- Working all hours. Better you should "get a life."
- Becoming complacent about your skills, position, or network.
- Believing any job is permanent.
- Being so self centered that you find yourself often working alone.

Conclusion

You know that change is coming. Be prepared. Train yourself, remain flexible, and network like mad. Treat the changes that do come as opportunities to see new vistas and, lastly, enjoy the journey.

Acknowledgment

This article would not have been possible without the assistance of Walter Cheney, SPHR, a true human-resource professional. His help in researching organizational laws was immensely rewarding.

References

Cordella, Antonio and Kai A. Simon. "The Impact of Informational Technology on Transaction and Coordination Cost," *Proceedings of IRIS 20*, 1997.

Gulick, Luther and Lyndall Urwick (editors) Papers on the Science of Administration, Institute of Public Administration, 1937.

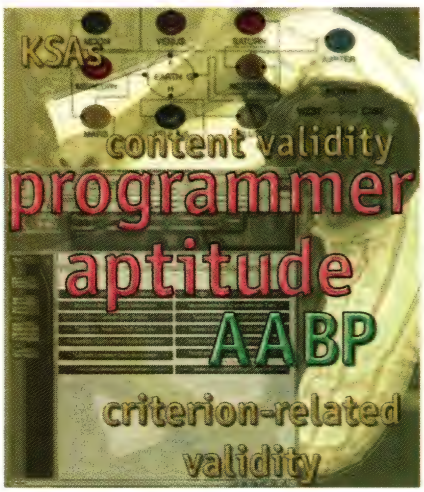
DDJ

The Walden Programmer Analyst Aptitude Test

Measuring programmer potential

Bruce Winrow

Because of the shortage of trained software developers, many companies rush to recruit and hire candidates, later discovering that a particular individual may not be the best person for the position. Since recruitment that ends up like this is very costly, organizations are looking for methods of assessing the true skills of potential employees. At Walden Personal Testing & Training (http://www.waldentesting.com/), we provide tools that help in the selection process. The tests we provide (which are typically licensed to and administered by an organization's human resource department) are not meant to replace the interview process, but are designed to be used in conjunction with in-depth interviews and detailed reference checking. Walden tests are neither I.Q. nor personality based. Generally, they are not multiple choice, but instead simulate what the job applicant is required to do on the job, as well as to determine who



is likely to be successful in technical training courses.

Initially, our most popular test was the "Aptitude Assessment Battery: Programming" (AABP) test, which was designed in the late 1960s by Dr. Jack Wolfe (a pioneer of early data processing and computer education) to evaluate computer programming potential. The five hour test is comprised of five very challenging problems. It has been, and continues to be, a successful evaluation tool. It not only answers the question, "Should you work as a programmer?" but also "How proficient will you be?" in a definitive manner.

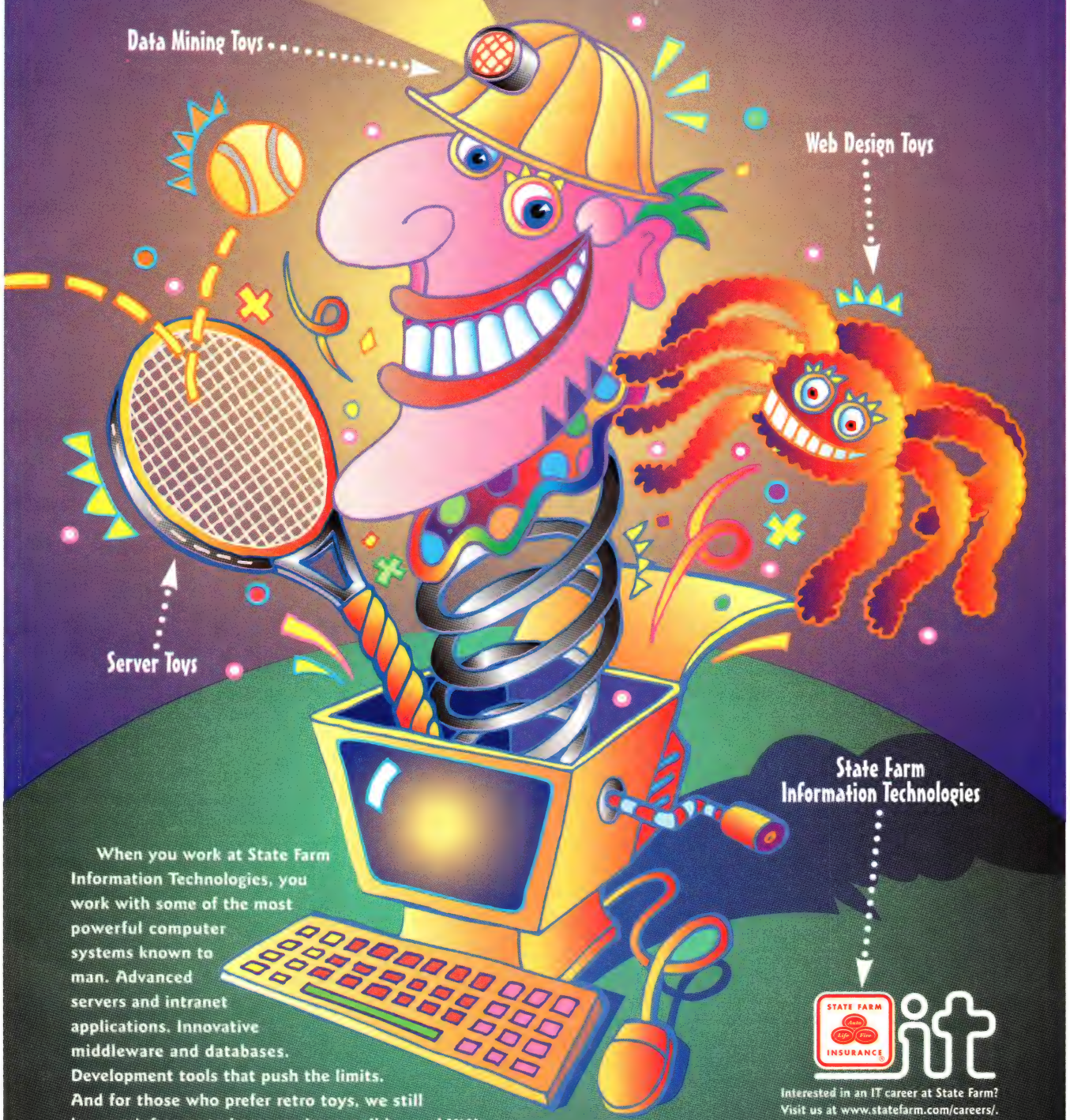
In 1983, we recognized there was a need to evaluate not only computer programming potential, but analytical skills as well. The ability to solve business-related problems and create, interpret, and dissect detailed programming specifications is an integral part of every developer's job. Programmers are no longer just coders, they're also expected to analyze detailed business requirements; hence, the job title "programmer analyst." This led to the design and implementation of a new test—the Programmer Analyst Aptitude Test (PAAT).

In developing PAAT, we made several design decisions, including:

- The test should evaluate aptitude for the work.
- The test should not exceed two hours.
- The test should incorporate several of the skills and abilities evaluated in earlier programming tests.
- At least 40 percent of the test should evaluate the candidate's analytical skills.

Bruce is vice president of consulting services for Walden Personnel Testing and Training Inc. He can be contacted at tests@waldentesting.com.

Come play with all the latest toys.



When you work at State Farm Information Technologies, you work with some of the most powerful computer systems known to man. Advanced servers and intranet applications. Innovative middleware and databases. Development tools that push the limits. And for those who prefer retro toys, we still have mainframes to keep you busy well beyond Y2K.

STATE FARM INSURANCE

it

Interested in an IT career at State Farm?
Visit us at www.statefarm.com/careers/.
Please refer to job code TOYS/DD
when e-mailing or faxing your resume.
E-mail: HRSF@STATEFARM.COM or FAX: 309-763-2831.
An Equal Opportunity Employer

State Farm Insurance Companies • Home Offices: Bloomington, Illinois

(continued from page S6)

- All questions should simulate programmer analyst job-related functions.
- The test must be designed so that the scoring is objective and suitable for future automation.
- The test must be field-tested and validated (content and predictive validation) to meet federal guidelines for fairness.
- The test must be suitable for both entry-level and experienced candidates and assume no prior experience or knowledge of information technology.

We have since added a one-hour version of the test that's available over the Internet, and is administered in booklet form as well.

How the Tests Were Developed

My first step in developing the PAAT, was to conduct a thorough job analysis of the programmer/analyst position. I started by soliciting current job descriptions from several client companies. It was apparent that approximately 10 job functions (tasks) were common to most of the descriptions received. Next, I identified essential knowledge, skills, and abilities (KSAs) for each of the job tasks. The challenge here was to incorporate those KSAs that were most often identified, incorporate several others that were tagged secondary traits, avoid multiple-choice questions, and limit the exam to two hours. We then decided that questions would be developed around the following six high-level abilities:

Programmers are no longer just coders, they're also expected to analyze detailed business requirements

- Follow a complete procedure involving multiple transactions.
- Create a routine procedure that solves a problem through the structuring of tasks.
- Interpret detailed specifications, follow rules, and incorporate symbols to solve problems.
- Analyze and execute a new instruction set, perform table look-up, and manipulate data to generate a solution.
- Analyze a complex business situation and search through voluminous data to understand and identify core requirements.
- Create a symbolic program to solve a business-related problem and then identify errors based on generated code.

Ultimately, we decided that the first two problems should be developed around common banking procedures, because many different types of transactions could be involved in the questions (deposits, withdrawals, and so on); the scenarios were common to most laymen, and business situations were involved. The actual creation of the questions then became easier, because all that had to be done was to model a realistic situation. Again, the trick was to limit both problems to approximately 25 minutes for the candidate and to ensure this represented more than sufficient time to solve the problems.

The third problem was developed around the general structure of computer machine language instructions (concepts of operation codes, data length, address, and so on). These concepts were made candidate friendly but required the ultimate test-taker to evaluate several complex expressions of increasing difficulty—all based on symbolic language architecture.

The fourth problem extended the concepts the candidate applied in problem three, but forced the individual to perform complicated table look-up procedures and then manipulate the data found to solve progressively difficult mathematical expressions. The creation of the question itself had as its foundation basic knowledge of tables and arrays, which are incorporated into most programming languages. The only challenge was to translate this theory into

PAAT Sample Question

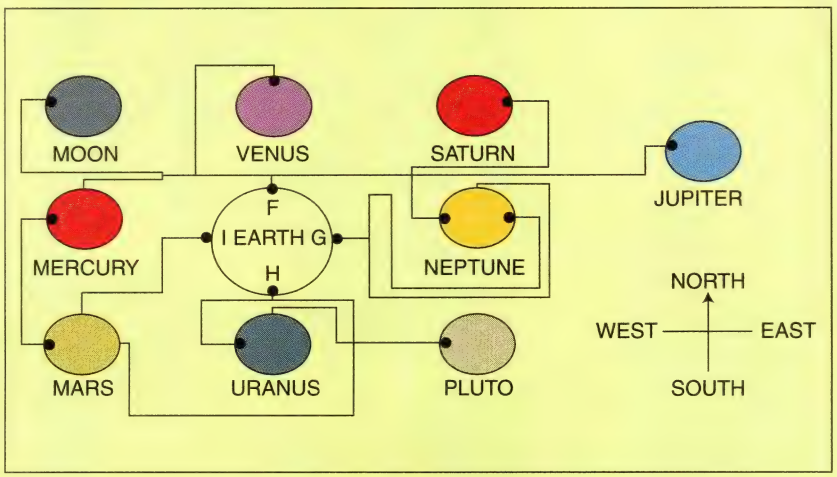
In analyzing problems, your logic ability and attention to detail are extremely important. In this question, you will be required to understand the symbolic coding of instructions necessary to logically direct the "Starfire" spaceship to its various destinations.

Rules for creating instructions from the coding symbols:

1. Every instruction must start with an S.
2. The next part of the instruction must be F, G, H, or I if Starfire is leaving Earth, or the number corresponding to the planet Starfire is departing from if it is not Earth.
3. The last part of every instruction must be the number of the planet destination plus "stop." However, if the code for the planet destination is an even number, it must be doubled before the "stop." Example: 32E for Venus destination.
4. The codes making up the middle of each instruction (between rules 2 and 3 above) are 1,2,3, and 4 in a combination necessary for Starfire to travel along the proper path.

Using our new "SYMBOLS," the instruction for sending Starfire from Earth (from the most northern space station) to Jupiter would be: SF131328E. (Please note that there is not enough information for you to actually solve this sample question.)

—B.W.



"100K, stock options and three-weeks paid vacation? Get back in line."

When you develop to Novell Directory Services® (NDS™) and NetWare® 5, you turn the tables on your future.

Employers know the global Internet economy is the future of business. If their applications don't play well with the e-commerce directory of choice, their applications won't play at all. That's why they're looking for savvy graduates who can develop to NDS—the cross-platform directory enabling the future of e-business.

Which means NDS development should be on your resume—after all, close to 50 million companies already use it, and it's employed by 80 percent of the Fortune 500. And if analysts are half-right, by the year 2000, directory services product revenues will hit five billion—that's five times 1998 revenues. What does that mean to the NDS developer class of 2000? Opportunity is going to beat your door down.

Join Novell's DeveloperNet® Academic Program and get NDS development skills through on-line training with Novell DeveloperNet University. Sign up for your future at http://developer.novell.com/education

Novell®

9 Novell, Inc. All rights reserved. Novell, NetWare, Novell Directory Services and DeveloperNet are ered trademarks, and NDS is a trademark of Novell, Inc. in the United States and other countries. All products and services mentioned are property of their respective holders.

(continued from page S8)
practice for an inexperienced candidate and restrict the time frame to about 20 minutes.

The fifth problem required detailing a complex business procedure in narrative form. It was modeled after a complicated billing system for a fictitious company and forced the candidate to analyze the specifications in order to correctly select options and calculate invoice totals. This was the only problem that incorporated multiple-choice questions and was done simply to minimize total testing time yet maximize the number of essential abilities evaluated.

The sixth and final problem actually had the candidate write a symbolic program based on an instruction set the individual learns from specifications provided at the beginning of the problem. The individual was also provided with coding in this fictitious language and must detect errors in the logic. Once again, the design of the question itself was based on concepts of operation codes and addresses and was built around a common business procedure—adding payments. The difficulty was ensuring that the entire problem could be completed within 30 minutes.

As a general rule of thumb, test problems were organized from easier to more difficult in the final version of PAAT. Table 1 lists the key traits evaluated by PAAT with an indication in which of the six questions they were contained.

Sample Question

The text box entitled "A Sample Question" presents an example of what a PAAT question might look like. You should be aware that there is insufficient data supplied in the problem to actually solve it. For reasons of security, we did not supply an actual question from the PAAT.

Based on the most recent candidates who have taken PAAT, the following statistics have been determined:

Population Tested	5322.00
Median (Mid-point)	74.00
Minimum	0.00
Maximum	100.00
Mean (Average)	64.64
Sample Standard Deviation	36.60

Testing and the Law

Antidiscrimination laws have been in effect in the United States since the passage of the Civil Rights Acts of 1866 and 1870, and the ratification of the 14th Amendment to the Constitution. More recently, there have been several acts and executive orders instituted to prohibit discrimination in employment. One of the most important of these laws was the Civil Rights Act of 1964.

	Prime Traits	Problem # in which the trait is evaluated
1.	Ability to think logically.	1, 2, 3, 4, 5
2.	Ability to follow procedural logic.	1, 5
3.	Ability to accurately solve problems	1, 2, 3, 4, 5, 6
4.	Ability to reason with symbols according to stated definitions.	3, 6
5.	Ability to perform basic arithmetic calculations.	1, 2, 3, 4, 5, 6
6.	Ability to understand and observe highly complex relationships.	3, 4, 5, 6
7.	Ability to work rapidly and accurately.	1, 2, 3, 4, 5, 6
8.	Ability to deduce generalizations from a series of specific cases.	3, 4
9.	Ability to follow instructions precisely.	1, 2, 3, 4, 5, 6
10.	Ability to analyze a problem not solvable by trail and error alone.	1, 3, 4, 5, 6
11.	Ability to pay attention to details.	1, 2, 3, 4, 5, 6
12.	Ability to adjust to changing situations.	3
13.	Ability to sustain concentration.	1, 2, 3, 4, 5, 6
14.	Ability to correct coded information.	6
15.	Ability to produce original ideas in solving problems.	5
16.	Ability to analyze charts, tables and graphs.	1, 4, 5
17.	Ability to persevere.	5
18.	Ability to reason abstractly.	5
19.	Ability to attack a problem from many angles.	5
20.	Ability to translate business specifications into symbolic logic.	6
21.	Ability to remember details/instructions.	1, 3, 4, 5, 6
22.	Ability to handle new problems.	1, 2, 3, 4, 5, 6
23.	Ability to solve problems utilizing reasoning with symbols.	3, 6
24.	Ability to read.	1, 2, 3, 4, 5, 6
25.	Ability to work independently.	1, 2, 3, 4, 5, 6
26.	Ability to work under pressure.	1, 2, 3, 4, 5, 6
27.	Ability to take corrective action.	6
28.	Ability to work at repetitive tasks.	1, 3, 4
29.	Ability to spot errors.	6
30.	Ability to comprehend written material.	1, 2, 3, 4, 5, 6
31.	Ability to remember plans, rules and overall policy.	5
32.	Ability to show endurance.	1, 3, 4
33.	Ability to create procedural logic.	2, 6
34.	Ability to learn.	3, 4
35.	Ability to comprehend quickly.	1, 3, 4, 6
36.	Ability to accurately check coded information.	3, 6
37.	Ability to think critically.	5, 6
38.	Ability to make decisions quickly and accurately.	1, 2, 3, 4, 5, 6
39.	Ability to show patience.	4, 5, 6
40.	Ability to sort data in a prescribed sequence.	5
41.	Ability to follow complex procedures.	5
42.	Ability to utilize program logic development techniques.	1, 2, 6
43.	Ability to interpret intricate business specifications.	5
44.	Ability to do a complete job.	1, 2, 3, 4, 5, 6
45.	Ability to analyze complex business procedures.	5

Table 1: *PAAT Prime Traits.*

Title VII of the Act prohibits discrimination because of race, color, religion, sex, or national origin, in all employment practices, including hiring, firing, promotional, compensation and other terms, privileges, and conditions of employment.

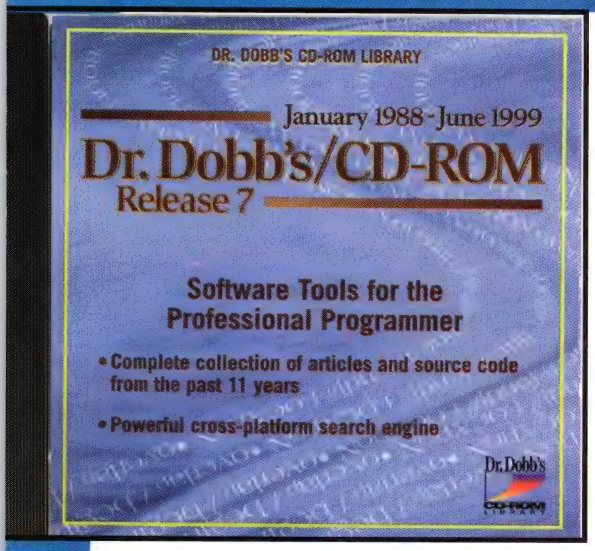
The U.S. Equal Employment Opportunity Commission (EEOC) was created to monitor companies and assure adherence to Title VII. The Equal Employment Act of 1972 provided EEOC with direct access to the courts and subsequently enhanced the Commission's effectiveness as a compliance-enforcing agency.

At first, the EEOC focused its efforts on the more blatant element of discrimination. Then the Commission addressed the application and use of employee selection procedures. The results of its actions were a set of standards for employee selections. These standards evolved into the *Uniform Guidelines of Employee Selection Procedures.*

The *Uniform Guidelines* are a set of standards to be used in determining whether employee section procedures are unlawful. The *Guidelines* were developed to be consistent with legal standards and validation standards accepted by the psychological profession.

The *Guidelines* apply to any procedure used to make an employment decision.

Expert advice when you need it most!



Dr. Dobb's/CD Release 7 contains over 11 years of cutting-edge computer information. Including over 132 issues of *Dr. Dobb's Journal* plus all the issues of *Dr. Dobb's Sourcebook* on one CD-ROM! Every Programming Paradigms, every Algorithm Alley, every column, every bit of source code from January '88 to June '99, all at your fingertips.

Dr. Dobb's/CD Release 7 brings together more than eleven years worth of perspectives and expertise from the most respected programmers in the industry. Since *Dr. Dobb's Journal* is written by programmers, for programmers you receive the news and views from inside the industry. *Dr. Dobb's/CD Release 7* offers both long time and first-time readers an invaluable programming resource.

Stop thumbing through back issues for the article you need. Avoid errors in re-keying source code, that you can copy and paste directly into your new programming project. Use *Dr. Dobb's/CD Release 7* for all your programming needs.

Release 7 provides you with all the cutting edge information from *DDJ* on: User Interfaces, Java Programming, Object Oriented Programming, Graphics Programming, Algorithms, Encryption and Compression, C/C++ Programming, Intelligent Agents, Client/Server Development, Patterns and Software Design, Software Debugging and Testing, File Formats, Embedded Systems, Database Development, Portability and Cross-Platform Development, and Web Development.

NEW RELEASE!

Price: $69.95

$29.95 Upgrade available to prior owners! To upgrade call: 800-822-1162

Fast, accurate and extensive, cross-platform search engine

HTML!

Call: 800-992-0549

International: 785-841-1631

Email: orders@mfi.com • Fax: 785-841-2624

Mail: Dr. Dobb's CD-ROM Library, 1601 West 23rd Street Suite 200, Lawrence, KS 66046-2703

Visit our web site for all Dr. Dobb's CD-ROM products and place your secure order at www.ddj.com/cdrom/

Dr. Dobb's CD-ROM LIBRARY

(continued from page S10)
This includes oral and written tests, interviews, review of application blanks, review of work experience, work samples, and physical requirements.

The underlying principle is that the employer policies or practices having adverse impact on employment opportunities of any race, sex, or ethnic group are illegal under Title VII, unless justified by business necessity. A selection procedure found to have no adverse impact generally does not violate Title VII. Adverse impact occurs when a selection procedure results in differential rates of selection or rejection of the various minority and gender groups.

This means that if an employer is using a selection procedure that does not have an adverse impact on a protected group, the employer may avoid the application of the *Guidelines*. If adverse impact exists, the *Guidelines* state that it must be justified on grounds of business necessity, that is, there is no other method of evaluating job suitability.

According to the *Guidelines*, validation demonstrates the job-relatedness of a selection procedure. For example, if a selection procedure is a test, the employer must show that the test scores are related to performance of the job for which the employees are being selected.

The *Guidelines* recognize three types of validation strategies:

- Criterion-related validity, which calls for the demonstration of a statistical relationship between selection-procedure scores (predictors) and the job performance of workers (criterion).
- Content validity, which calls for an investigation of the degree to which the test includes a representative sample of all the tasks that make up the job. The content of the selection procedure must be representative of important aspects of job performance.
- Construct validity, which calls for the demonstration that a selection procedure measures a specified attribute (construct), and that the selection procedure can be used to predict job performance. Construct validity requires a great deal of empirical data, and usually requires more than one study.

Of the three types of validation models accepted by the *Guidelines*, criterion related and content validity are the most practical and widely used.

We wanted to demonstrate that the test would predict the overall programming/ analytical potential of an applicant

Content Validity of PAAT

With regard to establishing content validity, all of the major essential functions (activities) required for the programmer analyst job position are identified based on data supplied by BCP Bank Card Processing Worldwide via a recent job description (1997). A copy of the job description for the programmer analyst is collected, verified as being up-to-date, and filed for future reference.

Each of the identified essential functions is ranked (weighted) in order of importance to the programmer analyst position by BCP Bank Card Processing Worldwide personnel.

Next, the traits, skills, or knowledge required to perform these functions are identified with an indication as to whether those essential traits are evaluated by a relevant PAAT question or some other means (such as on-the-job observation).

Also, the major secondary traits (nice to have, not essential) are identified for each of the key functions performed. It is noted whether those secondary traits are evaluated by a relevant PAAT test question or some other means. The goal is to ensure that all the major essential functions are specifically and accurately defined and categorized into behavior domains that can be evaluated.

For purposes of the programmer analyst, a complete job analysis was conducted. Fourteen key functions are identified for the job position. It is these 14 essential functions for which the PAAT is being used for evaluation.

Results of the Job Analysis Procedure

As the job analysis indicated, the programmer/analyst had to perform 14 essential functions to successfully execute the responsibilities associated with the job.

A total of 56 traits were judged to be essential to accomplishing the 14 essential functions. Of these essential traits, 36 were assessed by one or more sections of the test. Thus, a substantial portion (64.3 percent) of the intended domain was assessed by the test. In addition, 22 secondary traits (defined as those that are "nice but not necessary") were identified for the programmer analyst—Introductory/Trainee job position, of which five were tested, but not quantified. If the relative importance of the essential traits tested to the weighted importance of the job functions are considered, the overlap percentage improves to 69.3 percent.

Given the demonstrated relationship between the abilities and traits required to perform the 14 essential functions of the programmer analyst job, and those measured by the PAAT, the test represented a content-valid evaluation device for that job.

Predictive Validation

To conduct a predictive validation study of the Walden Programmer Analyst Aptitude Test, we wanted to demonstrate that the test would predict the overall programming and analytical potential of an applicant on the job as measured by supervisory ratings, for future employees in a programmer analyst job position.

In addition to the content validity described previously, BCP supplied us performance ratings of 63 applications programmers. The intent was to determine if a predictive relationship existed between

Knowledge-Based versus Aptitude Tests

Knowledge tests are used with the purpose of measuring the individual's existing knowledge that has been gained by actual experience or previous learning.

Aptitude tests are used in the expectation that they measure the capacity or potential of the individual to acquire and usefully use particular knowledge and/or skills.

Neither knowledge-based nor aptitude tests should be confused with I.Q. tests that produce a derived score (with a mean of 100) that is a measure of general mental ability.

—B.W.

success on-the-job and high scores of the PAAT. These scores were compared using the SPSS statistical software package.

There exists an empirical relationship, (r=0.457), which is significant to the 0.01 level, between high scores on the PAAT and programmer performance on-the-job. r is known as the "correlation coefficient" (or the "Pearson r"). It is used to indicate the strength (or lack thereof) of the correlation between two variables. On the PAAT, the conclusion was quite clear: The higher the candidate's score on the PAAT, the greater likelihood that the candidate would be rated higher on the job, based on performance. At the 0.01 significance level, you could expect this conclusion to have occurred by chance only one time out of 100. The legally accepted range for levels of significance is between 0.05 and 0.01. Given the empirical relationship between higher performance rankings and higher scores on the PAAT, the test represents a predictive valid selection device for this job.

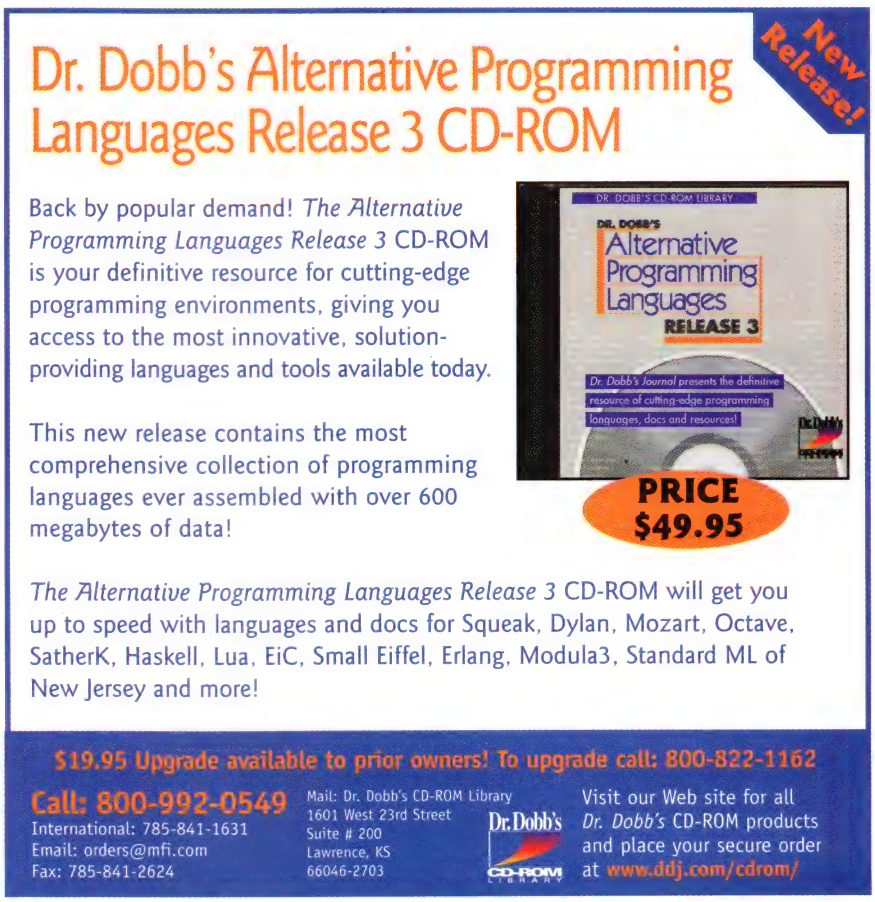
In addition (based on 63 hired candidates):

- The mean score for the PAAT was 82.2 percent.
- The median score for the PAAT was 90.0 percent.
- The mode score for the PAAT was 94.0 percent.
- The standard deviation for the PAAT was 12.4 percent.
- The range for the PAAT was 60 points.
- The minimum score for the PAAT was 40 percent.
- The maximum score for the PAAT was 100 percent.
- The mean score for males was 86.2 percent with a standard deviation of 11.8 percent.
- The mean score for females was 83.2 percent with a standard deviation of 13.6 percent.
- There was a significant empirical relationship for both males and females between high scores on the PAAT and high performance ratings on the job.

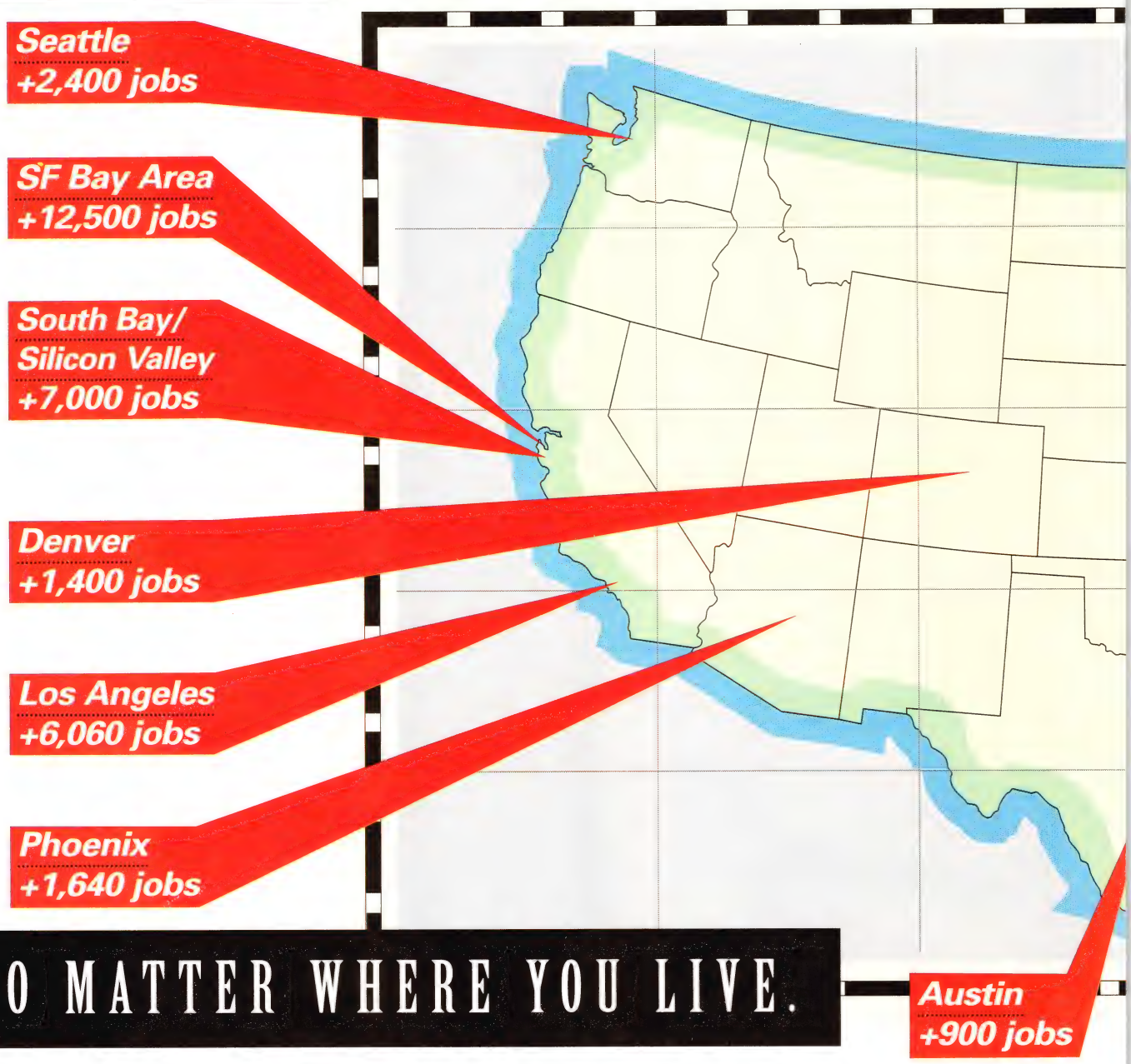
Conclusion

From a practical point of view, the PAAT definitely identifies those people with a high aptitude for programming and for systems analysis. It also identifies those individuals who are high-risk hires (low scores). This is the true value of the test. It minimizes the costs associated with a poor employment decision. A more recent development has been that an Internet version of a similar test is now available for clients who need an immediate response as to candidate suitability.

DDJ



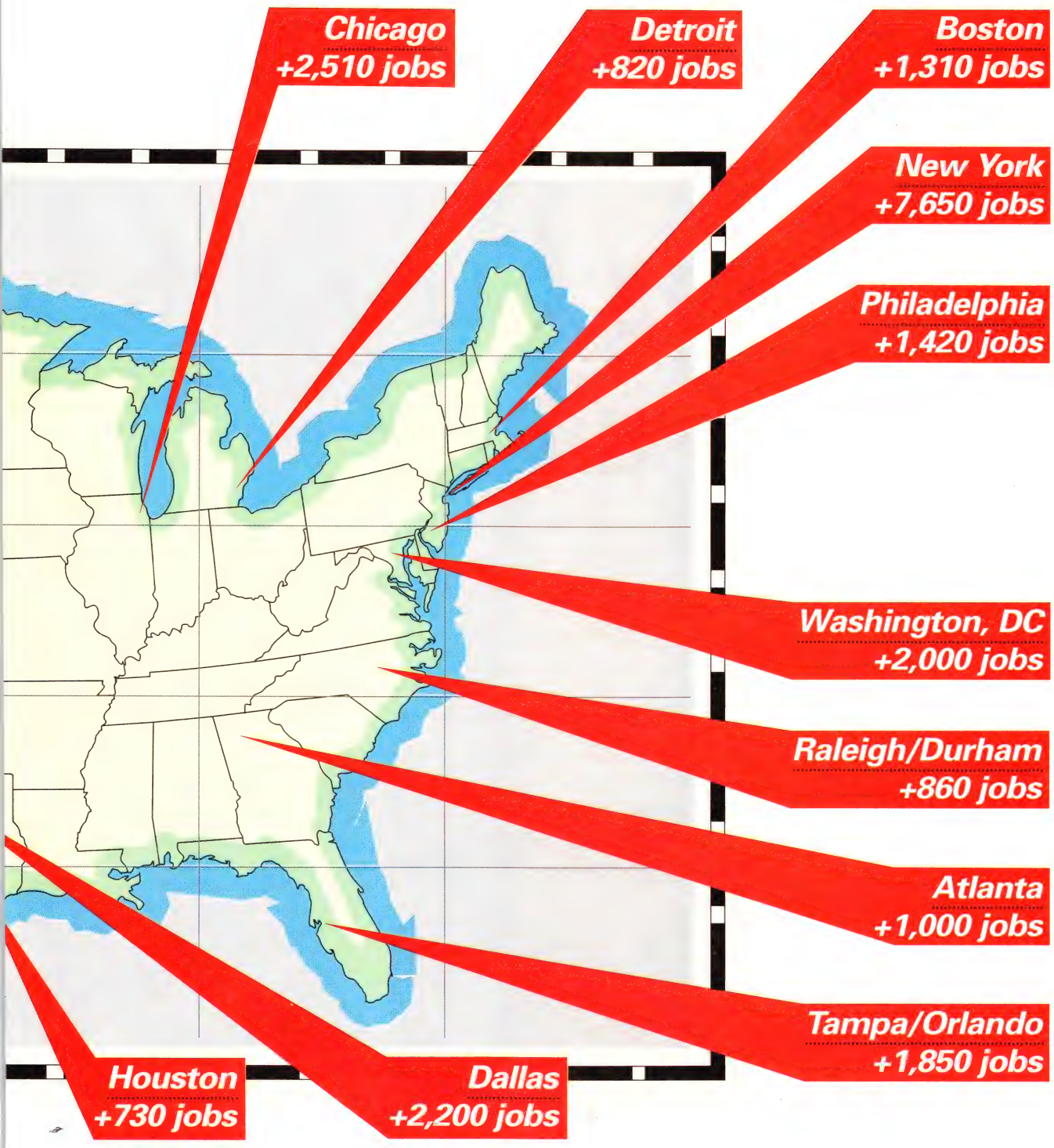
DICE HAS A HIGH TECH JOB FOR YOU,



NO MATTER WHERE YOU LIVE.

Looking for a high tech job? No matter where you live in the U.S., chances are that DICE lists a job that matches your skills and experience. Just go to www.dice.com and find your next job.

Maybe you want to move - or try a new place. Use DICE to locate your perfect job - you'll probably get a raise, too!



Better yet, use DICE's free JobTools to simplify your search - you can even have jobs matching your personal profile emailed directly to you! After you find your new job, use our relocation tools to ease the move.

No matter where you want to go, DICE is there. So go to www.dice.com and find your next job now!

dice
High tech jobs online

Real Jobs. Real Fast. DICE: 515-280-1144

Gauging Corporate Culture

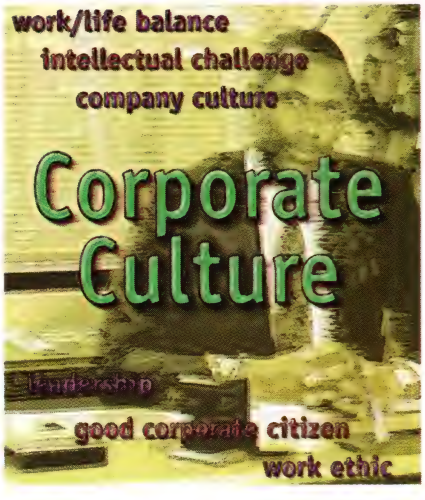
Matching your values to those of prospective employers

Greg Scileppi

Jerome Freeman was elated when he was offered a job as an applications developer at a promising startup that designs web sites devoted to e-commerce. But a month into the job, Freeman, a disciplined and independent self-starter, realized that the company wasn't meeting his expectations. While coworkers routinely worked 11-hour days, much of that time was spent in personal discussions in the coffee room while deadlines loomed. Meanwhile, Freeman was spending more and more time attending day-long meetings that consisted primarily of colleagues attempting to ingratiate themselves with senior management. Lacking a taste for office politics, he found these sessions even more frustrating since they left little time for him to spend actually designing applications.

Greg is executive director of RHI Consulting. He can be contacted at gscileppi@rhic.com.

It soon became apparent to Freeman that he had made a mistake in accepting the job. Worse, he wasn't precisely sure how he had so completely misread



what it would be like to work for the firm. Reluctantly, he began to look for another job.

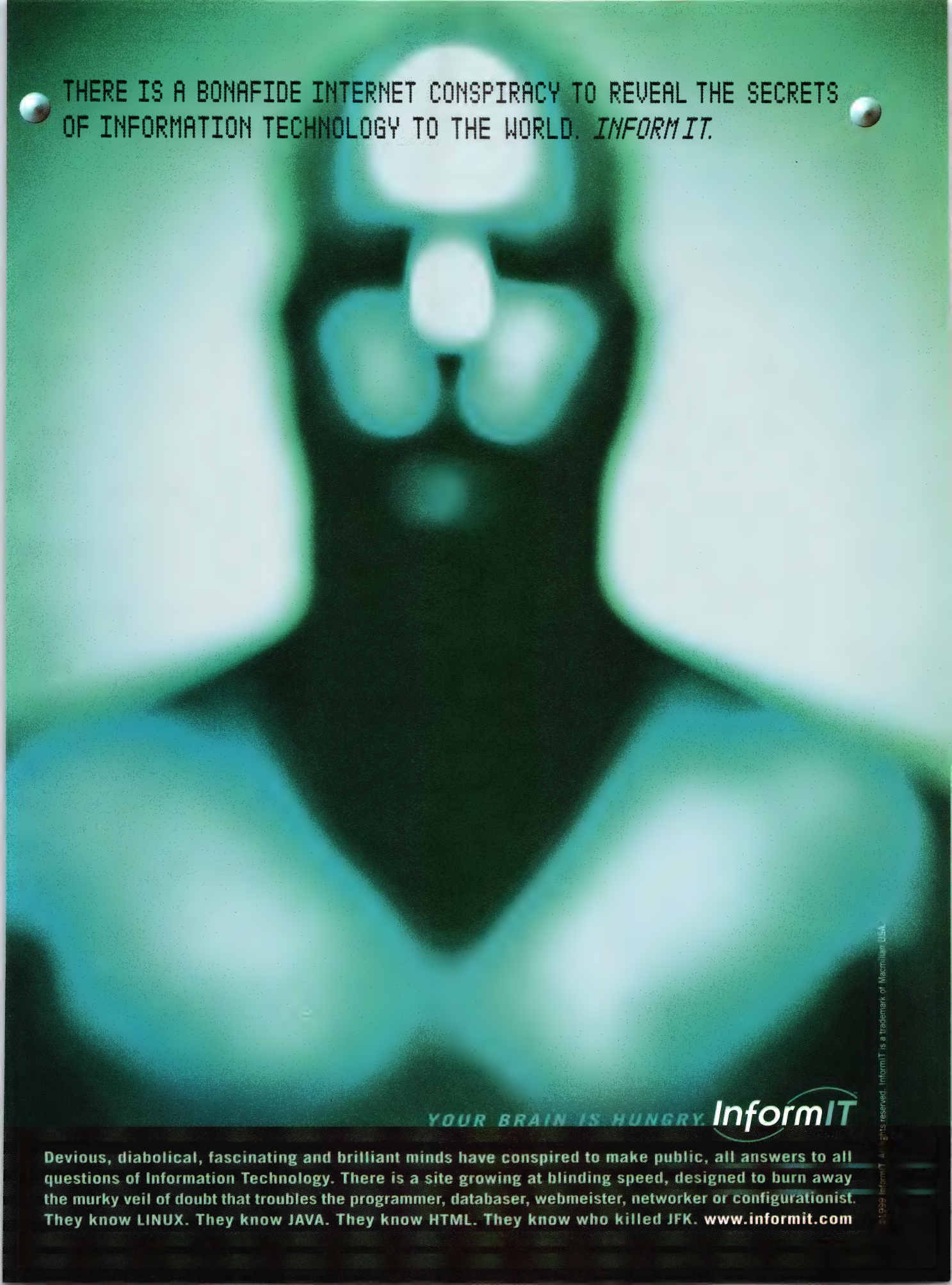
While Freeman's needs and preferences may not coincide with your own, they are reflective of concerns shared by a growing number of information technology professionals today as they assess their career options. A decade or two ago, technology professionals might have based their decision to sign on with a firm almost entirely on salary. But today, employees at all levels recognize that job satisfaction involves more than just a paycheck and an attractive benefits package. They realize their ultimate job satisfaction is also dependent on the unique set of values and attitudes displayed, sometimes subtly, sometimes not, by their employer. And, increasingly, they're able to get what they want.

Today's IT job candidates are in the driver's seat. And one of the factors they're using to evaluate a prospective employer is the organization's corporate values. The challenge, as Freeman learned firsthand, lies in accurately "reading" a firm and its work environment prior to making a full-time commitment.

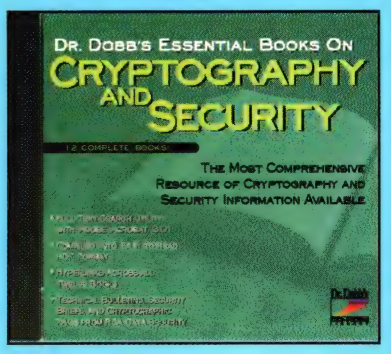
How do you make sure your values are closely aligned with the personality of your future employer? You can start by recognizing that there are many aspects to a company's culture. Just as an individual has a distinct personality he or she shows to the world, every business is imbued with its own unique set of values. For true job satisfaction, your task is to determine which values are most important to you and then find the firm that offers the closest match.

Work/Life Balance Number One for Many

Of all the corporate values held by U.S. businesses across all industries, probably the most widely publicized and most often cited by employees as important to


THERE IS A BONAFIDE INTERNET CONSPIRACY TO REVEAL THE SECRETS OF INFORMATION TECHNOLOGY TO THE WORLD. INFORM IT.
YOUR BRAIN IS HUNGRY. InformIT
Devious, diabolical, fascinating and brilliant minds have conspired to make public, all answers to all questions of Information Technology. There is a site growing at blinding speed, designed to burn away the murky veil of doubt that troubles the programmer, databaser, webmeister, networker or configurationist. They know LINUX. They know JAVA. They know HTML. They know who killed JFK. www.informit.com
©1999 InformIT All rights reserved. InformIT is a trademark of Macmillan USA.

Now in PDF!



Dr. Dobb's Essential Books on Cryptography and Security containts twelve essential books all programmers should have on their bookshelf! Written by leading cryptographic and security experts such as William Friedman, Bruce Schneier, Dorothy Denning, Alfred Menezes, and others, this CD-ROM is the definitive resource of cryptographic and security information.

The Essential Books on Cryptography and Security CD-ROM provides detailed documentation and analysis of: • Classical and Modern Encryption Algorithms • Private-Key and Public-Key Cryptography • The Data Encryption Standard • Signature Schemes (Digital Signature Standard) • Approaches to Key Distribution and Key Agreement Protocols • Block Ciphers and Stream Ciphers • Generation, Distribution, and Installation of Cryptographic Keys • Communication Architecture Encryption • Authentication Techniques • Applying Cryptography to Pin-Based Electronic Funds Transfer Systems • Measures of Secrecy for Cryptographic Systems • Access, Inference, and Information Controls • Plus Much, Much, More!

Price: $99.95
Text from 12 books

Dr. Dobb's CD-ROM LIBRARY

Call: **800-992-0549**
Email: orders@mfi.com / Fax: 785-841-2624
Mail: *Dr. Dobb's CD-ROM Library*
1601 West 23rd Street, Suite # 200
Lawrence, KS 66046-2703
Visit our Web site for all *Dr. Dobb's* CD-ROM products and place your secure order at

www.ddj.com/cdrom/

(continued from page S16)
them is a concern for workers' ability to reconcile the demands of their jobs with their personal lives.

Accommodations for telecommuting, flexible schedules, and dependent care are increasingly important as workers try to achieve a satisfying work/life balance, and a growing number of firms are now offering these benefits. A North Carolina manufacturer I work with, for example, arranges for school counselors to meet on-site with parents during regular working hours and gives employees time off to attend special events at the company's day care center. In a similar vein, an Oregon-based computer company recently extended its day care to include kindergarten and first grade.

Accommodations for telecommuting, flexible schedules, and dependent care are increasingly important

Intellectual Challenge Key to Job Satisfaction

Most IT professionals are aware of the need to keep their software and hardware skills current to maintain their marketability. In fact, a number of consultants come to me seeking contract work rather than full-time employment because the diverse assignments enable them to stay on top of a rapidly evolving industry. Technology professionals, in particular, view continuing education as a prerequisite to their career survival, an observation that was confirmed by a recent study conducted by RHI Consulting (the company I work for). When CIOs participating in the study were asked to choose the single factor, other than compensation, that helped managers retain IT talent, they placed ongoing technical skills training at the top of the list.

If learning is a priority for you, look for organizations that encourage their managers to solicit new ideas from employees at all levels. Firms that value intellectual growth are typically supportive of people who experiment with different approaches, and they tend to reward prudent risk taking. They invest generously in their staff by providing opportunities to learn on the job, including company mentoring programs and tuition reimbursement for outside training. Is it important for your future employer to offer you the chance to expand your skills and the freedom to move from project to project?

Effective Leadership

Most people would agree that the most effective leaders have a talent for building enthusiasm for an organization's goals and inspiring employees to put forth their best efforts. There is far less agreement, however, on what leadership means to each individual employee.

Some favor clear direction in their work, seeking well-defined, unambiguous tasks, for example, while others find this structure confining and prefer more autonomy. Then again, some employees may view too much autonomy as an abdication of leadership responsibilities.

Yet another group of employees, albeit a smaller group in today's fast-paced business environment, finds comfort in a traditional, hierarchical management model in which each position is defined explicitly. Their counterparts, on the other hand, look for leaders who will serve more as coaches than bosses, favoring project-driven collaboration over a chain of command. It's important to determine in which environment you feel most comfortable. In his case, Jerome Freeman realized too late that the culture at his new job was not a match for his personality and work ethic.

As you evaluate a prospective employer, consider its leadership. What are the characteristics of those in senior management? Try to identify the kind of behavior that is typically rewarded. A company's compensation program can also help you identify the predominant management style, with some companies tying bonuses to the success of departments or teams, and others basing them primarily on individual contributions.

Making a Difference

Is it meaningful to you for your firm to be seen as a good corporate citizen in the communities in which it does business? Grant-making programs and support for employee volunteer activities are powerful magnets for those who want to be associated with a company that is an active contributor to causes important to them. Such policies can also extend the reach of individual employees who want to support the work of specific organizations or to call attention to socially relevant issues. Employees at one Northern California network systems firm, for instance, have formed a special company-supported committee to organize activities such as weekend cleanups of local creeks and beaches, video conferences on environmental issues, and fund raising for environmental organizations.

Likewise, one Midwest clothing retailer offers employees up to 35 hours of paid time off per year to engage in community activities related to education or health and social services. Do any of these types of support reflect values you'd like your future employer to hold?

Company Culture

There are a number of ways to get an accurate feel for a business's values prior to signing on. While the job interview is the most obvious, there are steps you can take both before and after the interview to assess the underlying philosophy driving the policies of a potential employer.

For starters, the Internet is full of information about both large and small companies. Conduct a search to learn as much as you can about the firm before the interview, particularly as it relates to its business goals and industry reputation. The look and content of its web site can also give you a sense of the company's values and prevailing culture. Is the site cutting-edge and animated or subdued and sophisticated? This speaks volumes about the image they wish to convey to the business community and the public.

If you're called in for an interview, arrive a few minutes early so that you can observe how employees dress, how they interact, and what the physical workspace is like. Sometimes you'll have to look past the superficial to properly interpret the real messages an organization is sending. An offering of substantial on-premises services could reflect either a company eager to please its employees, or a corporate culture focused on maximizing the time employees spend on the job. One Silicon Valley firm, for example, provides such on-site services as dry cleaning, film developing, and dinners-to-go.

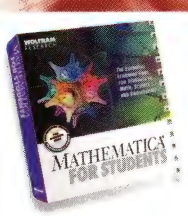
Grant-making programs and support for employee volunteer activities are powerful magnets

To get a further glimpse into an organization's values, speak with current and former employees. Professional trade association meetings can also provide an opportunity to connect with others who have knowledge of the company in which you are interested. In addition, you may want to speak with recruiting or consulting firms, which typically have strong ties within the business community and are familiar with the work environment and culture at local firms. If you're not successful obtaining enough information from these sources to make your decision, consider working as a contract or temporary employee before committing yourself to a full-time position.

A Winning Hand

You never really know a city until you've lived there. You never really know a neighborhood until you've moved in. So too with an employer. It's impossible to anticipate all of the subtle attitudes and policies that contribute to a company's culture. At the same time, by seeking to understand the firm's core values—then comparing them to your own—you can, at the very least, curtail any surprises and, at most, find yourself working for a company whose work environment and management style are ideally suited to your own personal and professional ambitions.

DDJ



for your math and science homework.
Get it at your nearest bookstore or visit:
www.wolfram.com/warp/ddj
NOW AVAILABLE VERSION 4
MATHEMATICA® FOR STUDENTS
WOLFRAM RESEARCH
© 1999 Wolfram Research, Inc. *Mathematica* and *MathLink* are registered trademarks of Wolfram Research, Inc. *Mathematica* is not associated with Mathematica Policy Research, Inc. or MathTech, Inc. All other product names mentioned are trademarks of their producers.

Careers in System Administration

The future looks bright for problem solvers

Barbara Dijker

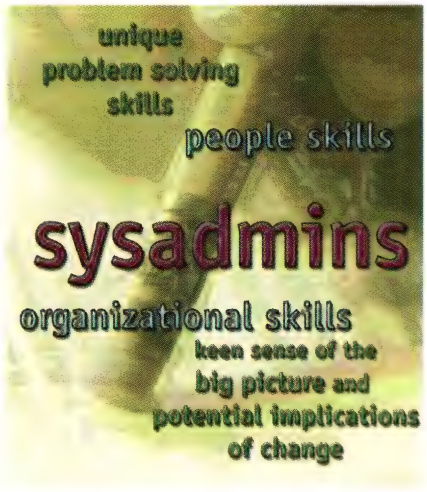
What are system administrators? In general, sysadmins are those people we complain to when our computer systems aren't working the way we expect. If they can make everything right, then they must be system administrators. Therefore, a system administrator is someone who solves problems in computer and network systems operations.

The problem set in computing and network operations generally includes all those system tasks users might want to offload—specification, evaluation, installation, configuration, integration, maintenance, data-integrity management, upgrade management, automation, security management, performance analysis, failure analysis, failure mitigation, recovery design, recovery implementation, testing, and more. In reality, the definition of system administration is usually a subset of all those tasks specific to the organization doing the hiring. For example, some organizations have a separate staff of network administrators to handle network administration, while in other organizations system administrators manage both computers and network infrastructure. Other organizations have separate staff dedicated to security administration. Some organizations allow users a high degree of self-administration that makes the system administrator's job focus more on the integration and systemic issues.

Barbara is vice president of the System Administrators' Guild (SAGE) and has been a system administrator for 13 years, working in aerospace, higher education, and the commercial sector. She can be contacted at barb@netrack.net.

Specialties and Career Advancement Paths

Because system administration is so broad, room exists for specialization. Examples of where you can specialize are security, data encryption, networks, heterogeneous integration, specific operating systems or platforms (Sun, NT, or Linux), specific subsystems (file sharing, mail, or naming systems), specific applications (supercomputing), and (more recently) web-server administration. Web-server administrators specialize in the operational issues specific to web servers, usually in the area of performance, security, reliability, and redundancy.



System administration is often thought of as a job in the trenches. However, good system administrators, through experience, hone unique problem solving skills, people skills, organizational skills, and a keen sense of the big picture and potential implications of change. These skills are invaluable for someone interested in a management track that can lead up the chain all the way to corporate information technology director and even chief technical officer.

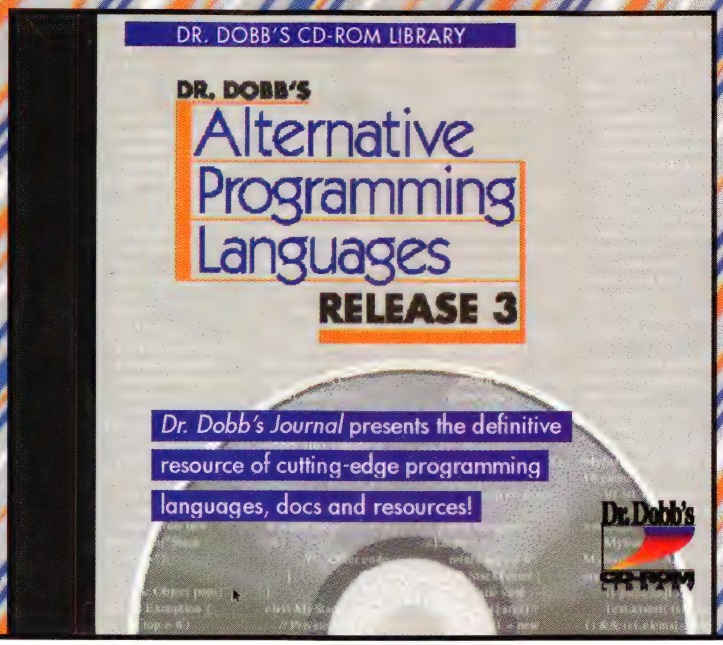
Because system administrators seem to enjoy the challenge of actually fixing problems, career advancement often tends to stay on a more technical track. System administrators have plenty of responsibility for the operational integrity of the systems they administer, so the lure of responsibility for staff or budgets is less appealing. Advancement technically can be toward:

- Consulting, which often provides the ability to select work or truly focus on a specialty.
- Back-line, away from the "front-line" of interruptions to the more challenging problems.
- Specialties, focused and more detailed work, more available in larger organizations.

Large organizations may provide predefined advancement from the front-line of basic user support or "help-desk" through multiple levels of system administration toward the back-line of elite troubleshooters. Some companies even have a "special forces" team of system administrators to swoop in and fix really tough problems.

Market Areas and Industries

Every organization that has computing systems needs someone to perform system administration tasks. While every such site may have that need, solving it does not always involve hiring a full-time system administrator. When the load of system administration tasks is low, they are usually delegated to an existing staff person whose primary responsibility is something else. Interestingly, this is often the genesis of a system administrator. That person finds they enjoy the system administration tasks more than his other work.



Dr. Dobb's Alternative Programming Languages Release 3 CD-ROM

Back by popular demand! *The Alternative Programming Languages Release 3* CD-ROM is your definitive resource for cutting-edge programming environments, giving you access to the most innovative, solution–providing languages and tools available today. This new release contains the most comprehensive collection of programming languages ever assembled with over 600 megabytes of data!

The Alternative Programming Languages Release 3 CD-ROM will get you up to speed with languages and docs for Squeak, Dylan, Mozart, Octave, SatherK, Haskell, Lua, EiC, Small Eiffel, Erlang, Modula3, Standard ML of New Jersey and more!

New Release!
PRICE! $49.95

CALL: 800-273-9208 US and Canada

Visit our website: www.ddj.com/cdrom/

E-MAIL: orders@mfi.com
FAX: 785-841-2624
INT'L: Use mail, fax, e-mail, or call 785-841-1631

Dr.Dobb's CD-ROM LIBRARY

$19.95 UPGRADE AVAILABLE TO PRIOR OWNERS! TO UPGRADE CALL: 800-822-1162

MAIL ORDERS:
DR. DOBB'S CD-ROM LIBRARY
1601 WEST 23RD STREET, SUITE 200
LAWRENCE, KS 66046-2703 USA

(continued from page S20)

The affect of this on the market for system administrators is that you can find a job in system administration in any industry in any market sector. Small, growing, and changing organizations feed the free-lance consulting and contract system administrator markets. Other organizations feed the consulting and contract pools with experienced system administrators who are looking for a change. In short, system administrators work for everything from churches and hospitals to multinational and Fortune 500 companies.

Market Demands and Trends

Some people have long predicted that the need for system administration would die out like dinosaurs as computers and applications become more automated and sophisticated and solve their own problems. The opposite is actually the case. As computers and applications have become more sophisticated, they've created different problems from those they may have solved. For example, ubiquitous computer networks have solved many problems in transport of information and resource sharing. However, for each problem solved, networking has introduced at least a dozen in the areas of security and data integrity. As problems crop up, system administrators have to figure out what to do about them to allow users to continue the "real" work.

There has never been a market glut of systems administrators. In fact, there are probably five jobs available for every existing sysadmin today. In fact, most experienced system administrators find they are doing the work of five people. Demand has out-paced supply for at least the last six years.

Of course, this is mostly subjective information. There are at least two problems with clearly quantifying trends in the system administrator job market. The first problem is that most system administrators have a job title that has nothing to do with system administration — a member of technical staff, systems programmer, systems analyst, and so on. According to studies conducted by the Systems Administrators' Guild (SAGE; http://www.usenix.org/sage/jobs/salary_survey/index.html), fewer than half of all system administrators are given title recognition for it. The second problem is that market analysts don't usually track system administrator as a unique job title because of the first problem. Any surveys of system administrators are based on self-identification, not job title.

Salaries and Salary Trends

As you would expect with increasing demand and shortage of qualified applicants, salaries for system administrators are attractive and competitive. According to SAGE studies, the average salary for all system administrators — regardless of experience, industry, or geography — was approximately $62,000 in 1998. System administrators with 1-2 years of experience on average earned $47,000 while those with 10 or more years of experience earned more than $70,000. About five percent of all sysadmins earned over $100,000 in 1998. As with most jobs, salaries are higher in the commercial sector than they are in education, government, and other organizations. Salaries are also higher in parts of the country where there is more demand and higher costs of living, such as California and New England.

Every organization that has computing systems needs someone to perform system administration tasks

Salary increases for system administrators tend to be larger than those in other jobs, usually to keep someone from leaving for greener pastures. The increase in average salary from 1997 to 1998 was 6 percent. Salary disparities between men and women are significantly less in system administration than other fields. In fact, the 1998 survey referenced here indicates women made about 15 percent more than men. However, that could be attributed to more men coming into the field than women. System administration tends to attract more women than other technical and computing fields. Sixteen percent of system administrators are women.

Consulting and Contracting Versus Direct Employment

Because of the large demand for system administrators, there is also a large market for "on-demand" system administrators through contract employment or consulting. Some system administrators prefer consulting or contracting because it provides greater variety and flexibility.

System administrators who are independent consultants tend to be specialists and make significantly more money than those who work for a firm and are contracted out. (For more information, see the SANS Salary Survey, http://www.sans.org/newlook/publications/1998salary-survey.htm). As with most other computing contract employment, those who work for contract firms tend to make slightly more money than those who are directly employed. In this respect, system administration is similar to other technical fields. The only difference may be that due to significant demands for system administrators, making a go of it as an independent may be more readily lucrative than in other fields. There are plenty of opportunities regardless of which employment structure is most attractive.

Job Requirements

Becoming a system administrator is relatively straightforward. Anyone curious to explore a challenge while possessing a reasonably solid background in computing fundamentals may find themselves drafted anyway. Due to demand, organizations are also more likely to consider offering entry-level positions with training than in other computing fields.

Gaining a background in computing fundamentals is pretty easy. A degree in computer science isn't required, just a basic understanding of how computers, operating systems, and applications work and interact, logic, binary numbers, a little electronics, and some common sense. The most important traits of a new system administrator are the abilities to find and digest information, solve problems, develop creative strategies, and recognize and work within one's own limits. System administrators who have formal education and a degree tend to be preferred and sometimes required for jobs. However, that's not necessarily because the education is particularly relevant to the job, it's more because it is evidence of follow through and perseverance. Many practicing system administrators hold degrees in technical fields other than computer science, such as physics.

System administrators solve problems daily. The problems change every day and there is no step-by-step guide for solving them. So a system administrator has to be able to think on his/her feet, learn about the aspects of the problem, find the cause, find a fix, and then implement one that actually works within the constraints of the situation. Sysadmins are usually working on production systems, because it is expensive to duplicate a system purely for the system administrator's benefit. Consequently, it is imperative that a system administrator knows when he/she is in over his/her head — and how *not* to create more problems. Other than that, it's all

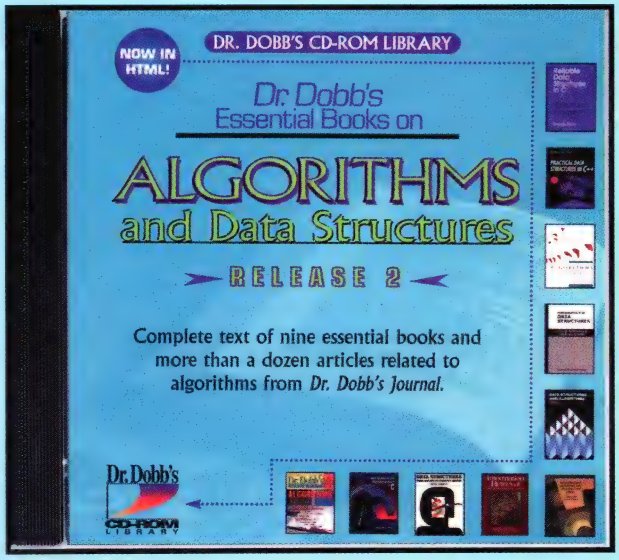
Now In HTML!

Full text of 9 essential books and more than a dozen articles from DDJ!

Dr. Dobb's Essential Books on

Algorithms and Data Structures

Release 2 CD-ROM



There's nothing more fundamental to programming than algorithms and data structures. Your CD-ROM includes nine of the most important books ever written for programmers about the subject.

These books give you proven and trusted ways to analyze and solve problems in C, C++, and other languages. Take advantage of the research and testing that went into developing the algorithms and data structures included on this CD-ROM!

Master practical programming techniques. Ensure that your programs are understandable, maintainable, and efficient—even as they grow in size. The right data structure can make your operations simple and efficient, the wrong one can make an operation cumbersome.

Learn from the experts. Set up your algorithms the right way the first time—before you ever write that first line of code!

Upgrade for previous owners
800-587-8319 $29.95.

ncludes These 9 Essential Books

- *ata Structures: From Arrays to Priority Queues* by Wayne Amsbury
- *ntroduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest
- *nformation Retrieval: Data Structures & Algorithms* edited by William B. Frakes nd Ricardo Baeza-Yates
- *undamentals of Data Structures* by Ellis Horowitz and Sartaj Sahni
- *eliable Data Structures in C* by Thomas Plum
- *ata Structures and Algorithm Analysis in C* by Mark Allen Weiss
- *ractical Data Structures in C++* by Bryan Flamig
- *ata Structures, Algorithms and Program Style Using C* by James F. Korsh and eonard J. Garrett
- *ata Structures and Algorithms* by Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman

Plus more than a dozen articles related to algorithms from *Dr. Dobb's Journal!*

Buy your CD-ROM today for only

$69.95!

Visit the *Dr. Dobb's CD-ROM Library* website:

www.ddj.com/cdrom/

MAIL ORDERS:
DR. DOBB'S CD-ROM LIBRARY
1601 WEST 23RD STREET, SUITE 200
LAWRENCE, KS 66046-2703 USA

Dr. Dobb's CD-ROM LIBRARY

(continued from page S22)
about experience and system-specific—UNIX, NT, applications—training.

Training and Education

There is no widely accepted formal curriculum for system administration. Unfortunately you can't go to the local university and sign up for a degree program in system administration.

Still, there are plenty of education and training opportunities for system administrators, including commercial education programs, university continuing education, vendor-specific courses, conference and seminar tutorials, and self-education through books. Commercial technical training companies, such as Technology Exchange Company (http://www.technology-exchange.com/) and Global Knowledge Network (http://www.globalknowledge.com/), also provide system-administration courses. Major platform vendors (Sun, Microsoft, Novell, and the like) provide system-administration training as well.

Nonetheless, universities are starting to develop system-administration curricula. The University of Indiana, for instance, offers an independent study program in system administration called "UNIX System Administration Independent Learning" (USAIL; http://www.uwsg.indiana.edu/usail/). The System Administrators' Guild (SAGE; http://www.usenix.org/sage/) provides publications, conferences, and tutorial programs on system administration. The number of books on system administration increases all the time. There are currently about 75 books in print on various aspects of system administration. Two books stand out as required reading for UNIX system administrators: *UNIX System Administration Handbook*, by Nemeth, Hein, Snyder, and Seabass (http://www.admin.com/) and *Essential System Administration*, by Frisch (http://www.ora.com/catalog/esa2/).

System administration is often thought of as a job in the trenches

Furthermore, there are certification programs for system administration. Most existing certification programs are vendor specific; such as Solaris and/or NT administration, for instance. The Linux Professional Institute (http://www.lpi.org/) is developing certification for Linux professionals, who are essentially system administrators. SAGE is conducting an occupational analysis of system administration to begin the process of the only platform-independent system administration certification program.

Conclusion

The profession of system administration and programs for system administrators are growing all the time. In part, to meet this demand, SAGE is currently conducting a pilot program in mentoring and plans to have an established mentoring program next year. Furthermore, companies such as Sun are creating new multifaceted programs to spawn and encourage new system administrators. Why? because system administration is a career that is in high demand—with no let up in sight.

DDJ

27 fathoms.
And an uncharted world.

But not for long. Every day, Motorola is unlocking mysteries, exploding myths and exposing the world to exciting new ways to work, play and live. In fact, there is virtually nowhere on Earth where our technological influence hasn't reached.

Want to check your e-mail from your car? How about surfing the Net on your pager? Or downloading some MP3 tunes to your cell phone? It's all possible and you can make it happen at Motorola. We have openings in the suburban areas of ***Atlanta, Austin, Boston, Chicago, Ft. Lauderdale, Piscataway, Phoenix, Montreal, San Diego, Toronto, and Vancouver****. Start charting an amazing future for yourself in the following:*

TECHNOLOGY/ENGINEERING OPENINGS

- ***Analog Design Engineers***
- ***ASIC Design Engineers***
- ***Digital Design Engineers***
- ***DSP Engineers***
- ***Information Technology Professionals***
- ***Oracle DBAs***
- ***Satellite Communication Engineers***
- ***Software Test Engineers***
- ***Software Tools Engineers***
- ***Software Real Time/ Embedded Engineers***

At Motorola, you can chart an exciting career path while enjoying exceptional compensation and benefits all along the way. For consideration please submit your resume on our website: ***employment.motorola.com (please reference Dr. Dobb's Journal)*** *Motorola is an Equal Opportunity/ Affirmative Action Employer.*

Visit our website at:
employment.motorola.com

MOTOROLA

XML and CORBA

The XML/IT toolkit combines the best of two worlds

Dirk Hamstra

The eXtended Markup Language (XML) provides great flexibility when it comes to data description and data structure. Applications need more than just data, however. Services are key application components. In this article, I'll discuss how XML, in combination with the Common Object Request Broker Architecture (CORBA), can provide an application-development environment that is flexible and rich in services.

The key benefits CORBA provides are the separation of interface and implementation and the abstraction of common application services. Services—security, management, transactions, and the like—are lifted out of the application logic and implemented in separate processes. XML, on the other hand, focuses on data description and structure. XML is actively used in such areas as data interchange, cross-format document management, stream-based representation of complex objects, and semantic bridging.

Although XML is not a replacement for either CORBA or COM+, some people still view it as infinitely flexible middleware. Consequently, there are a number of common misconceptions about XML.

- XML is not middleware. XML deals with the representation and structure of data. CORBA is a distributed object standard. Data representation is an important but small part of a distributed application.
- XML is not an ORB. This is mostly "marketingware." One XML-vendor, in fact, promotes its XML-based software as a replacement for an ORB. Upon close inspection, however, they use XML in a role similar to Common Data Representation (CDR) in Internet Inter-ORB Protocol (IIOP).
- XML and HTTP are not the same. A protocol must fit the usage profile of an application. Some parts of an application may require a synchronous connection while other parts are better served using a queuing or multicast mechanism. Ideally, message delivery is based on message properties that are set and resolved at run time. XML can flow over any transport mechanism. This means that XML data can be shipped over protocols such as MQ or IIOP, in addition to HTTP.
- XML does not handle all data types. All data in XML is represented as strings. This helps human readability, but reduces performance for nonstring data types. In general, XML's type system is primitive, which makes it useful on the Web, but not suitable for manipulating complex, typed data. Through the combination of Interface Development Language (IDL) and IIOP, CORBA provides a variety of data types (integer, string, octet, sequence, and the like).
- XML is not a better IDL. For all practical purposes, interfaces can be defined in any arbitrary language. Replacing IDL with XML is just a swap of languages. If you compare Listing One (the interface in XML) with Listing Two (the interface in IDL), for instance, you'll see that trading IDL for XML is not useful, since XML is not as compact and precise as IDL.

Because CORBA and XML are complementary technologies, there are a number of benefits you realize by combining the two technologies, such as:

- Replacement of CDR. XML overlaps with CDR. Replacing CDR with XML allows for more flexibility in the representation of data above and beyond the data types currently available. XML can also reduce problems with IIOP versioning.
- XML+IDL. Using XML and IDL in combination provides for the discovery of object structure and methods at run time, rather than generating compilable code. Benefits include economical, efficient, and extensible requests and responses for dynamic object creation and method invocation.
- Object serialization. XML is a good language to stream out objects. XML is close to a bytestream. Shipping flattened XML objects using a CORBA infrastructure is a natural match.
- Rich data. A CORBA call can return an XML document with, for example, monitoring statistics, including object references in the data. This allows more flexible and comprehensive interaction between software processes.

CORBA+XML: An Example

In general, using CORBA and XML together means that the input and output of a CORBA method call is expressed in XML. To illustrate this process, I'll use an XML formatting that's based on the functionality of a Java tool called XML|IT from CareFlow (http://www.careflow.com/). XML|IT allows automatic tagging of the results returned from calls to CORBA-based services. It also includes utilities that support conversion of XML-tagged documents to Java structures, and vice versa.

In addition to assuming the existence of CORBA-based back-end services, XML|IT assumes the use of IDL-to-Java compilers to generate Java stubs and skeletons, and uses CORBA's Dynamic Invocation Interface standards specification of CORBA. The main client program routine (*DIICall*) can be embedded inside other Java-based clients, CGI scripts, or Java servlets. For example, to convert a Java structure to XML, you would call XML|IT's JavaToXML from within Java using Listing Three. As a result, an XML representation of *bankObj* will be put in the *String* called *xml*. Mapping arrays to XML is handled through the use of a dimension attribute. Listing Four is the Java representation, while Listing Five is the converted XML representation.

For the conversion to and from Java, XML|IT makes use of the Fascade Pattern combined with Java reflection. The

Dirk works at Vitria Technology Inc. and can be reached at dhamstra@vitria.com.

Get on the fast track to Windows 2000 development.

There are no shortcuts to greatness. But some tracks are faster than others. To engineer the next generation of high-powered applications for Windows® 2000 and the Web, these Microsoft® learning resources for the Visual Studio® 6.0 development suite are your fast track–*your edge.*

Gear up with the in-depth training in the *Microsoft Mastering* series, *Deluxe Learning Editions,* and *MCSD Training Kits.* And stoke your performance with the expert guidance of *Programming Applications with Microsoft Windows,* Fourth Edition, and other books in the *Microsoft Programming* series.

The developer learning resources for Visual Studio 6.0. The ultimate performance enhancers. And the fast track to Windows 2000 readiness. **Made by Microsoft.**

Where do you want to go today?

Microsoft®

mspress.microsoft.com

© 1999 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Press, Visual Basic, Visual C++, Visual Studio, *Where do you want to go today?*, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

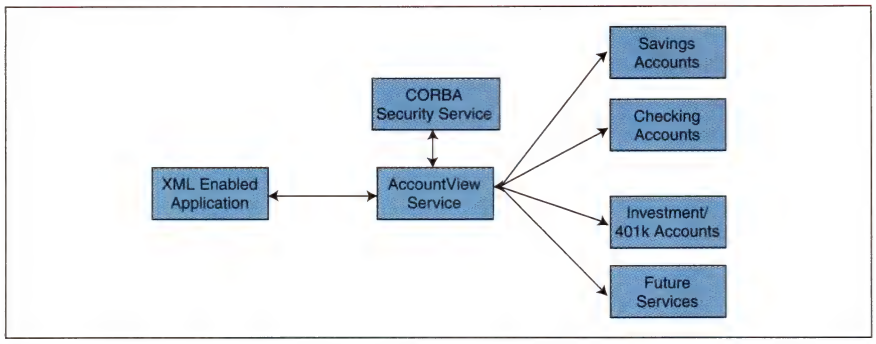


Figure 1: *Generic view of a system that implements the* accountList *function.*

(continued from page 98)

Interface *com.careflow.xml.Accessor* implements the Fascade Pattern. The *Accessor* provides a unified interface to elements that may have as a parent an *Array*, a plain structure, or no object at all. The class *com.careflow.xml.ReflectionTraverser* iterates over all the elements of an *Object* by reflection. Each element is visited by the given implementation of *ReflectionVisitor*. During a *ReflectionTraversal*, the current node may be a field in a structure, an element in an array, or the root element. The *Accessor* provides a common interface to these three cases, allowing them to be conveniently handled by the *ReflectionVisitor* and the *ReflectionTraverser*.

To invoke a method on a CORBA service, the input XML document must be formatted using some general guidelines. Listing Six exemplifies this. The XML document can be divided into three sections.

- The first part is contained between the <servicename> tags and describes the location and name of the CORBA service that will be used. Listing Six shows how to make a dynamic invocation. Static calls do not need to reference the implementation repository.
- A section detailing the method invocation follows the service specification.
- The last part delineated by <returntype> specifies the data type containing the results of the invocation.

Listing Six shows a CORBA method invocation expressed in XML, specifically the invocation of the method *accountList* on a service called *Bank*, located on machine *ChicoSpuugbeest*. The function takes two parameters, *customerId* and *accountNo*. The result of the operation will be returned in a sequence called *CustomerAccountSequence*.

Figure 1 provides a generic view of a system that implements the *accountList* function. The function is implemented by a CORBA service called *accountView*. The associated IDL for the service can be expressed as in Listing Seven. How the *accountView* service is implemented is hidden from the requestor. Execution of the *accountList* operation may involve multiple transactions to different systems to build the list of accounts. Also, because account information can be sensitive, a security service may be used to authenticate and authorize the request. The transport medium for XML is also open. The request could come from a browser that passes the XML flow, or from a message queue to the CORBA service.

Conclusion

CORBA and XML are different, complementary pieces of the same puzzle. CORBA provides a solid distributed framework with rich application services. XML gives you a flexible, dynamic data representation and structure. XML is intended for the storage and manipulation of text comprising human-readable documents such as web pages, while architectures such as CORBA tie together cooperating computer applications exchanging data that will probably never be read by anyone. Neither of these technologies will replace the other, but instead they will increasingly be used together.

DDJ

Listing One

```
<module>
    <name> banking </name>
    <interface>
        <name> savings_account </name>
        <inherits> bank_account </inherits>
        <operation>
            <name> deposit </name>
            <param>
                <name> amount </name>
                <type> money </type>
                <direction> in </direction>
                        </param>
        </operation>
    </interface>
</module>
```

Listing Two

```
module banking
{
        interface savings_account : bank_account
                {
            void deposit (in money amount);
        };
};
```

Listing Three

```
//given Object bankObj
JavaToXML translator=new JavaToXML();
String xml=translator.toXML(bankObj)
```

Listing Four

```
String[][] accountInfo=new String[1][1];
accountinfo[0][0]="Red R Hood";
System.out.println(xml.toXML(accountInfo));
```

Listing Five

```
<java.lang.String dimension=2 length=1>
    <item length=1>
        <item> Red R Hood </item>
    </item>
</java.lang.String>
```

Listing Six

```
<servicename>:\ChicoSpuugBeest:ddjdem:::IR:Bank</servicename>
<method>accountList</method>
<argument>
     <arg flag=in>
        <customerId>123456ABD</customerId>
     </arg>
     <arg flag=in>
             <accountNo> * </accountNo>
     </arg>
</argument>
<returntype>CustomerAccountSeq</returntype>
```

Listing Seven

```
interface accountView
    {
        exception reject {
            string reason; // error text};
               attribute string userId;
               attribute string password;
               attribute string customerId;
               attribute float accountNo;

               account accountLogin (in userId, inpassword)
               raises (reject, invalidLogon);
               account accountList (in customerId, in accountNo,
                   out sequence CustomerAccountSequence)
                   raises (reject, invalidCustomerID);
    };
```

DDJ

ImageGear®99:

Building powerful imaging solutions is now as easy as child's play

The most powerful imaging toolkit now offers several new Plug-Ins with support for **10 CAD/Vector formats**, *high-speed* **ISIS® scanning**, **OCR**, **Barcodes** *and more.*

100+ Formats!

ImageGear 99 Overview:

Standard Formats
AFX, Animated GIF, ASCII, ATT, AVI, BMP, BRK, CALS, CCITT Group 3-2D, CCITT Group III, CCITT Group IV, CLP, CUT, DCX, DIB, EPS*, EXIF, FlashPix, GIF, GX2, ICO, IFF/ILBM, IGF, IMARA, IMG-Xerox, IMG-GEM, IMT, IOCA, JFIF, JPEG, KFX, Lossless JPEG, LV, MAC, MO:DCA, MSP, NCR, OS/2 Bitmap, PBM, PCD, PCX, PGM, PICT*, PNG, PNM, PPM, Progressive JPEG, PSD, RAS, RLE, SGI, SUN, TGA, TIFF, Transparent GIF, Winfax, WMF, WPG*, XBM, XPM, XWD

Special Options:
DICOM, ABIC, JBIG, Wavelets,...

*Raster only.
License required from Unisys for formats using LZW compression.

NEW! CAD/Vector Plug-In:
DWG, DXF, DGN, HPGL, CGM, HPGL/2, ME10, DRW, RIS, PICT

NEW! Raster Plus Plug-In:
30 extra raster formats

IMAGEGEAR®99 Remember those classic wooden ABC blocks growing up … how easy it was to build anything our hearts desired! AccuSoft remembers and has updated its own building blocks. ImageGear 99 offers numerous technology choices on a simple plug-in basis, so your solutions can be developed as easily as your "younger" creations.

ImageGear 99 provides many powerful imaging technologies while allowing you to still maintain low level control of your applications. These options will help you to increase the market potential for your existing products and make new application development a breeze. Our comprehensive line of imaging products include ActiveX, DLL, Java class libraries, Java Beans, and sophisticated Web server solutions.

Let AccuSoft be your source and partner for imaging technology. Our strengths can be leveraged in many ways from toolkits to custom solutions. We want to show you how to improve your application development by building value one block at a time.

Imaging Technology for 25 function categories:

- High-speed image display
- Image loading and saving
- High quality image printing
- Advanced TWAIN scanning
- Sophisticated image processing
- Highly optimized image data access
- Graphical user interface components
- Individual and batch format conversion
- Highly optimized compression and decompression
- Easy to implement animation
- Photographic special effects
- Multiple image transition effects
- Advanced Internet support (URL, FTP, etc.)
- Native CMYK color space support
- Database integration support
- Color conversion/separation for many color spaces
- Advanced arbitrary ROI support
- ImageClean automatic image correction
- Best annotation support on the market
- High-speed scanning based on ISIS® standard *(Plug-in)*
- OCR support *(Plug-in)*
- Premium 1D and 2D barcode technology *(Plug-in)*
- Check imaging ABIC format support *(Plug-in)*
- Import of 10 CAD/vector formats *(Plug-in)*
- Special format support - over 30 available

ISIS scanning ImageGear PLUG-IN | OCR ImageGear PLUG-IN | BARCODE ImageGear PLUG-IN | IMAGE CLEAN ImageGear PLUG-IN | ART ImageGear PLUG-IN | Check Imaging ImageGear PLUG-IN | RASTER PLUS ImageGear PLUG-IN | CAD ImageGear PLUG-IN

Other AccuSoft imaging products: ImageGear Java - Pure Java imaging toolkit ■ **NetVue** - The most powerful web imaging technology for OEMs, Integrators, VARs, and IT professionals. **Winner of the *"Best of AIIM'99"* award** in the *"Web based document management"* category ■ **DocVue** - A document imaging component with many advanced features ■ **Medical Imaging SDK** - Advanced DICOM support ■ **DICOM Communications SDK** - Easy to use DICOM protocol toolkit ■ **ExamiNet** - Web plug-in for medical imaging

© AccuSoft Corporation. All Rights Reserved.
AccuSoft and ImageGear are registered trademarks of AccuSoft Corporation. High Performance Imaging and NetVue are trademarks of AccuSoft Corporation. All company and brand names are trademarks or registered trademarks of their owners.

ISIS is a registered trademark of Pixel Translations, a division of Input Software Inc.

AccuSoft®
High Performance Imaging™

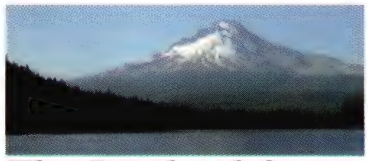
Two Westborough Business Park • Westborough, MA 01581 • Tel (800) 741-7769, (508) 898-2770 • Fax (508) 898-9662. **For more information, demos and evals visit our website at www.accusoft.com**

DIGITAL IMAGING INTERNET IMAGING MEDICAL IMAGING

It's not a Workstation without PGI Compilers and Tools.

Test drive The Portland Group compilers and tools on your Intel processor-based Linux workstation. Be among the first 500 readers to register and check the *Dr. Dobb's Journal* box on the PGI web page to receive a 15-day evaluation package and a 20% discount off your first PGI product purchase. PGI's *PGF77*®, *PGF90*™, *PGHPF*® High Performance Fortran, *PGCC*® ANSI C and *PGC++*™ compilers feature auto-parallelization, full native OpenMP user-directed parallelization, and all of the common extensions you expect in workstation compilers. The *PGPROF*® profiler and *PGDBG*® debugger allow you to develop and tune parallel applications for Intel processor-based workstations, servers and clusters.

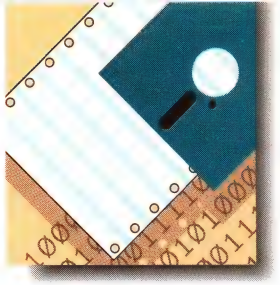
Industry-leading Fortran performance • Designed for scientists and engineers • Fast compile times • 64-bit integers and pointers
Byte-swapping I/O • DEC/Cray/IBM extensions • Interoperable with gcc, g77 and gdb • OpenMP directives/pragmas • Loop unrolling
Pentium III SSE support • Cache tiling • HTML documentation • Uniform development environment across Linux, Solaris and NT
Graphical debugger and profiler • No parallel runtime fees of any kind • 30 day money-back guarantee



The Portland Group

(503) 682-2806 FAX: (503) 682-2637 e-mail: sales@pgroup.com
www.pgroup.com

PGI, PGF90 and PGC++ are trademarks and PGF77, PGHPF, PGCC, PGPROF and PGDBG are registered trademarks of The Portland Group, Inc. Other brands and names are the property of their respective owners.



Magic

Michael Swaine

Any sufficiently advanced technology, Arthur C. Clarke said, looks like magic. The rate of technological advance, the loose interpreters of Gordon Moore say, doubles every 18 months. Multiply Clarke by Moore and the conclusion is inescapable—a day will come when *current* technology looks like magic.

The premise of this month's column is that that day is upon us. The magic moment was reached in physics much earlier, of course. It came early in the century when, trusting their math and their data more than their common sense, physicists accepted that the fundamental nature of the world is not only different than anyone thought but different than anyone can think. The quantum moment.

Technology is humbler than quantum physics, though, and only occasionally does it take a leap so large that the mass of people can't see the connection between the new technology and older, accepted (even if not really understood) technology. The telephone or telegraph may have been one such magic technology, but radio was not—it was originally known as "the wireless," defined in terms of existing technology. It's like something you already know, but with waves in place of wires. Television was radio with pictures. Movies were moving pictures; that is, defined in terms of photography. New technologies may be startling when you first see them, amazing that whatever it is could even be done, but usually they are understood as extensions of known technologies. Some technologies, like lasers, seem to come from nowhere, and seem a bit more magical.

Now, though, the pace has picked up. Computers are the reason. Children who in a past era would have been taught to use particular tools are now being taught how to build tools. Steve Wozniak has been quoted as saying that computer science is the most important discipline that children can be taught. As a computer scientist, one is a little embarrassed by such grandiose claims, but is there any more powerful tool to place in anyone's hands than a knowledge of programming?

It's a tool that's being handed to a lot of kids. This has been happening, at an increasing pace, for long enough for generations of programmers to have made contributions to the advancement of technology. No wonder the pace has picked up. No wonder it's starting to look like magic. But that's my premise, and I haven't supported it yet. Maybe the following observations will prove the point.

Michael is editor-at-large for DDJ. *He can be contacted at mswaine@swaine.com.*

Attaining Invisibility

The goal of MIT's Oxygen project (described in *Scientific American*, August 1999, and http://www.lcs.mit.edu/anniv/) is to push computer and information technology beyond what passes for ease-of-use today, making it not only accessible to everyone, but richly exploitable by everyone. Michael Dertouzos, director of MIT's Laboratory for Computer Science, envisions a world in which the technology we now access by typing and staring at screens in boxes on desks will be accessed via speech directed to the walls or to a tricorder-like device that everyone will carry. Thirty MIT faculty are involved in the five-year project to develop Oxygen, which Dertouzos hopes will become as pervasive as the air. (In the speech in which he laid out the Oxygen project, Dertouzos predicted a 300-fold increase in human productivity in the next century. In the *Scientific American* article, this has been scaled back to a modest 300 percent increase. I'm pretty sure Dertouzos meant 300-fold, or 30,000 percent.)

Oxygen comprises eight technologies: a handheld device called Handy 21, an in-wall or in-car-trunk device called Enviro 21, a network for collaboration called Net 21, speech recognition, intelligent information retrieval, tools that allow anyone to script the devices and automate repetitive tasks, intelligent and informed tools for collaboration, and the ability to customize the technology to your own needs. One of the features of the current technological landscape that is not included in the vision is shrinkwrapped software. Apps will go away, Dertouzos predicts.

Dertouzos's eight technologies are technologies that we would probably all agree are desirable, and that are being worked on in places other than MIT already. But what makes them special, and makes the Oxygen project interesting, is Dertouzos's claim that these are the only important technologies. Actually, it's just the last five technologies that he makes this claim about. He says:

> The five technologies of speech (and other perceptual capabilities), knowledge access, automation, collaboration and customization are the only new kids on the block. Out of the thousands of things that we can imagine doing in the new world of information, these five are the foundations on which any new activities that help us do more by doing less will be built.

Doing more by doing less. It's Dertouzos's mantra. And when it works, it's magic.

Quicker Than the Eye

One technology crucial to the Oxygen project is the Raw microchip, the project of Anant Agarwal of MIT. The idea is to get rid of the traditional interface to the microprocessor—the instruction set—and let the software have access to the raw hardware.

The Raw chip is an array of identical tiles, each containing memory and function units, and each connected to its neighbors by wires. Excess logic gates at the junction between tile and wire make up a programmable switch. The (software) compiler can program those switches to direct the flow of signals through the chip. Among other things, the software can create registers as needed, storing data in a convenient spot to break up long wires that would incur long delays.

Agarwal's group has built a Raw simulator and written a compiler to take advantage of it. In a test application, the compiler achieved a 10-fold speed increase over any existing conventional processor. Students tweaked the code for a further 10-fold increase. Agarwal hopes to see compilers developed that get that full 100-fold speedup when the chip becomes a real device.

Doing the Impossible

> ...trying to find a computer simulation of physics, seems to me to be an excellent program to follow out...and I'm not happy with all the analyses that go with just the classical theory, because NATURE ISN'T CLASSICAL, dammit, and if you want to make a simulation of nature, you'd better MAKE IT QUANTUM MECHANICAL, and by golly it's a wonderful problem because it doesn't look so easy.
>
> —Richard Feynman (1981) (*International Journal of Theoretical Physics*, v.21, p. 486)

The open source movement invades all corners of computer science. Now, open

source has moved into quantum space with the open source quantum computing project. The goal of the project is to simulate a quantum computer on a conventional one. The web site for the project (http:// www.openqubit.org/) has become a portal for quantum computing work.

Actual code has been released recently. There are now in fact several quantum computer simulators, and the field has become so crowded that it requires reviewers. A review of all existing simulations at press time has been posted at http://www.dcs.ex.ac.uk/~jwallace/simrevab.htm. Quantum computing— quick refresher here— promises to surpass the theoretical limits of conventional computing by taking advantage of the quantum nature of reality. In a conventional computer, bits take on one of two discrete states.

In a quantum computer, quantum bit— or qubits (pronounced "cue bits")— exist in a quantum superposition of states when not being measured (measurement collapses each of them to some discrete state).

This indeterminacy is the source of the computational power of quantum computers, which can be seen by considering memory registers built of qubits. A quantum register of size n requires 2 to the n complex numbers to represent its state, so it can store 2 to the n complex values, while a conventional register of the same size requires (and can store) just n integers. The storage capacity of quantum registers thus increases exponentially with size. That's where the magic lies.

Problems that can be solved in polynomial time are generally thought of as being tractable, while problems that can only be solved in exponential time are thought of as intractable. Quantum computing, by virtue of this exponential expansion of storage capacity, promises to make the intractable tractable.

In practice, this is exploited through quantum parallelism. The idea is to apply a function to a quantum register, thereby exploiting the exponential capacity of the register to get a lot of calculations simultaneously. Exactly how you exploit this capacity has been a bit of a puzzle. The problem is that you are essentially looking at the outputs corresponding to all possible combinations of inputs. The trick, then, is to make the output of all the possible inputs turn out to be something interesting, and useful. Peter Shor and L.K Grover, among others, have come up with algorithms that do just that.

Is This Your Number?

It was Shor's algorithm for factoring large numbers, developed at Bell Labs in 1994, that brought about a significant shift in the perceived seriousness of the field of quantum computing. Shor's algorithm, if implemented on a quantum computer, would crack many encryption schemes in polynomial time, making them worthless.

This is what most of the quantum simulation work has focused on— simulating Shor's quantum algorithm for factoring large numbers.

It turns out that the hard part of factoring a large number can be reduced to finding the period of a particular periodic function involving the number. Shor's algorithm finds the period of that function.

The number to be factored is n; x is an integer coprime to n, the period of the function is represented by r, and the function itself is *x to the a mod n*.

The algorithm uses two quantum registers. It loads register 1 with an equally weighted superposition of all integers from 0 to q–1, where q is a power of 2 between n squared and n squared times 2. It calls the contents of register *1 a*. Then it applies the transformation *x to the a mod n* to register 1 and stores the result— a superposition of all outputs— in register 2. Due to quantum parallelism this takes only one step. It then measures the second register, observing some value k. The superposition of output values has collapsed into the value k. Quantum magic also has

INFORMATION TECHNOLOGY • KAISER PERMANENTE • WWW.KP.ORG/JOBS

Advancing technology to improve health care.

That's the edge you give our team.

And their patients.

Over 9,000,000 members across the country turn to Kaiser Permanente's more than 90,000 physicians, nurses, allied health and business professionals for expertise in managing their health care needs. Over 3,000 experienced Kaiser Permanente IT professionals utilize state-of-the-art technologies to facilitate the efficient delivery of this high quality care.

Using the latest tools on an unparalleled spectrum of applications, while enjoying the advantages of ongoing training and education, our IT professionals support clinical innovation, elevate member satisfaction and maximize efficiencies. They are leading the next generation information age in health care.

Bring your expertise to Kaiser Permanente today in one of the following disciplines. Become involved in exciting initiatives such as Web development, extranet, intranet, enterprise data security solutions, network planning, client/server-based projects, network infrastructure, project management and technology planning.

Position Locations:
California, Colorado, Georgia, Ohio, Oregon, Maryland, and Washington D.C.

PC LAN
NETWORK ENGINEERING
DATA SECURITY
DATABASE ADMINISTRATION
DATA ANALYSIS
DATA WAREHOUSING/DATA MODELING
PROGRAMMING/ANALYSIS
TECHNOLOGY RESEARCH
SYSTEMS PROGRAMMING/ SYSTEMS ADMINISTRATION
BUSINESS ANALYSIS/ PROJECT MANAGEMENT
WEB DEVELOPMENT
SOFTWARE ENGINEERING/ SYSTEMS ARCHITECTURE
TELECOMMUNICATIONS
QA TESTING

Find out more about Kaiser Permanente Information Technology by visiting www.kp.org/jobs and clicking on the "IT Division" tab. Submit one resume only, indicating **Job code: PAD-DDJ119**, preferably by e-mail to: kpitjobs@kp.org, or fax to: 1-800-839-5140. The tangible and intangible benefits of working for an organization of our size and scope will change your career perspective in a healthy way. Affirmative Action/Equal Opportunity Employer.

KAISER PERMANENTE

Software Careers | THE ULTIMATE JOB RESOURCE FOR THE PROFESSIONAL DEVELOPER

www.developercareers.com

www.developercareers.com

Software Careers | THE ULTIMATE JOB RESOURCE FOR THE PROFESSIONAL DEVELOPER

Veri Cool Software

Veri Cool Careers

Seagate Software NSMG and Telebackup Systems, Inc. have merged with VERITAS Software to create one VERI COOL Company (NASDAQ:VRTS)!

Our company, doing business under the VERITAS name, is now the LARGEST independent developer of storage management software solutions that protect, access and manage data from the Desktop to the Data Center.

Join us and you'll get a chance to work on some of the world's coolest integrated software products for Windows NT, Novell NetWare and Sun UNIX environments. Recognize any of these award-winning products...Backup Exec, WinInstall, Volume Manager or Cluster Server?

We're on the move and growing, employing over 2,300 people in 17 countries worldwide. Our East Coast locations are Boston, MA; Takoma Park, MD; Durham, NC and Orlando, FL. In the West, check us out in Boulder, CO; Roseville, MN; Mountain View and San Luis Obispo, CA.

www.veritas.com/jobs

Immediate Opportunities

SOFTWARE ENGINEERING

Application/GUI (C++, NT/UNIX)

UNIX Kernel

NT File Systems

NW Protocol

QA Software Test

Tech Writer

TECHNICAL SUPPORT

Sr. UNIX Systems Analysts

Sr. NT Systems Analysts

NT or NW Help Desk

IS

UNIX System Administrator

Oracle DBA

Director of Application Development

Sr. LAN/WAN Architect

If you qualify for one of these positions, we'll reward your accomplishments with a highly competitive compensation package and great benefits which include Stock Options, ESPP, Matched 401(k), Comprehensive Insurance and Tuition Reimbursement.

Right about now, you might be thinking—"Sounds like a 'hot' company—so how do I apply for one of these 'cool' jobs?" Here's how:

- **E-mail: veritas@webhire.com**
- **Fax (800) 245-3197. For immediate consideration, reference SC1199 on fax cover sheet/e-mail subject line.**

VERITAS Software is an EOE with a corporate commitment to diversity.

VERITAS

Atomic Games Inc

Atomic Games is now hiring!

All levels of game programming positions available.
Lead, Sr., and Jr. game programming positions.
Game System and AI programming positions.
2D and 3D DirectX graphics programming,
Visual Studio and C/C++. Mac experience a plus.

Atomic is looking for 3D artists with real-time PC game experience.
3DMAX experience required.
Also hiring 2D artists. If you are a wizard at Adobe Photoshop, we are looking for you.

Other positions are available.
Re-location to Houston, TX is required.
Please send resume to:

www.atomic.com/jobs.html
resume@atomic.com

Great jobs.

No hassle. No cost. No problem.

You deserve a better job. And now it's easy to find one. Actually, we'll find one for you. Our service is absolutely free, absolutely confidential, and we pursue only the opportunities you personally choose. In fact, we don't even send your resume to a prospective employer until you approve them. Fill out a Member Profile — we do the rest. This amazing opportunity is brought to you by **Career Central for Developers**, the world's easiest-to-use high-technology search solution. Please visit us at **www.careercentral.com/dev** and complete a Member Profile today. The sooner you do, the sooner we'll find you the job you really deserve.

Career Central *for* Developers

CONNECTIONS USA, Inc., Ft. Lauderdale, FL, designs, implements, and maintains Internet-based software products. We are looking for experienced Programmers, System Admins and Graphic Artists.

Technical Skills Required

rogrammers
C (5-10 yrs), one high level language other than C, one assembly language, UNIX (5-10 yrs), BSD of some variety. HTML & CGI a plus.

ystem Administrators
UNIX admin exp, exp PC hardware, Windows 95 config & use. One real language (ie: C or Java), one interpreted language (ie: PERL, AWK, TCL). HTML & CGI a plus.

raphic Artists
Exp. w/ standard graphics prog. for creation of Web age graphics, banners, & ads. Animation exp a plus.

Casual Work Environment
Flexible Hours

Compensation Commensurate with Experience

www.connectionsusa.com
For immediate consideration, contact
ONNECTIONS USA, Inc. by Phone: 954-525-0300, ext. 71
by Fax: 954-525-1122
by e-mail: Shelley@connectionsusa.com

CONNECTIONS USA, Inc. is an Equal Opportunity Employer

---

http://www.scientific.com

Professional Software Developers
Looking for career opportunities?
Check out our website!
Nationwide Service
Employment Assistance
Resume Help
Marketability Assessment
Never a fee

Scientific Placement, Inc.

800-231-5920 800-757-9003 (Fax)
das@scientific.com

SPECIALIZING IN HIGH-TECH SEARCH AND RECRUITING FOR 14 YEARS

RICCIONE & ASSOCIATES, INC.

972.380.6432 (V)
972.407.0659 (F)
hitec@riccione.com

Full-Time and Contract Positions:

- Win NT/95/98
- UNIX
- Real-Time
- Applications
- Systems-Level
- Firmware
- SQA/Test
- Product Mgmt.
- SW Mgmt.
- Telecom
- Datacom
- Vendor SW
- Internet
- Multimedia
- Control Systems

VISIT OUR WEB SITE

RICCIONE ON-LINE COMPUSEARCH
WWW.RICCIONE.COM

Software Engineers
Nationwide Opportunities - Never a Fee

e Have Excellent Positions at:
Software Companies - Major *to* Start-ups
International Financial (Wall Street) Firms
Corporate R&D Centers
Major Industrial Companies

ills Sought:
C++ and C on UNIX or Windows
Java Development
GUI Development Visual C++/MFC
COM/DCOM, CORBA, ODBC
Oracle, Sybase, SQL Development

mpetitive Pay:
$60 to 150k depending on skill and COL, many positions have bonus and stock options

HAMILTON
TECHNICAL PERSONNEL, INC.
P.O. Box 369 West Hurley, New York 12491
ll free - 888-497-8462 - *FAX 914-679-5704*
email: dobbsjobs@hamiltontech.com
http://www.hamiltontech.com

The Job Dr. ®
Computer & Software Recruiting

- Games 3D
- Software Developers
- VB/SQL Developers
- Visual Basic/ VC++
- Java, ActiveX, MFC
- Oracle, SQL, Delphi
- Programmers

Salaries Range $65K-120K
CALL, FAX OR E-MAIL TODAY!

1-888-81-JobDr
Fax 949-360-1808
find@thejobdr.com
www.thejobdr.com
Aliso Viejo, CA

BATEMAN INC.
We Understand The Programmer's Mind℠

HARD-CORE WINDOWS, NT, JAVA AND LINUX DEVELOPMENT JOBS.
• • •
THE COOLEST COMPANIES ACROSS THE U.S. ALL THE TIME.

FIND YOUR NEXT GIG AT
WWW.BATEMANINC.COM

711 Daily Drive, Ste.106 • Camarillo, CA 93010
TEL: 805-383-3338 FAX: 805-383-3337

This space comes with over 300,000 professional programmers.

Call today to make it your space:
Elizabeth Doherty
Software Careers
650-655-4193

Software Careers

This isn't a game!

From the world's first color TV sets to weather tracking systems to the international space station, we don't just make things, we make history. Located on the waterfront in Camden, N.J. (across from Philadelphia), **L-3 Communications** produces high tech communication systems for commercial and military applications. You'll find the flexibility and challenges you're looking for in a career.

Software Engineers

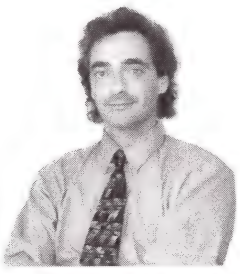
All positions require a BS/EE or BS/CS (or comparable technical degree) and experience with workstation-based engineering tools. US citizenship and background investigation required. We promise an exhilarating atmosphere, top clients and fascinating top-secret programs. We also offer highly competitive salaries and outstanding comprehensive benefits packages including medical, dental and vision.

Mail: L-3 Communication Systems-East
Attn: PAS-SWC
1 Federal Street, A&E Bldg.2C
Camden, NJ 08102
Fax: 609.338.3720
E-mail:
patricia.a.spruill@L-3Com.com
Website:
www.L-3com.com
EOE.

L3 communications
Engineers engineering life.

---

careermag.com
A very human resource.



It worked for me!

www.careermag.com

Software Careers

THE ULTIMATE JOB RESOURCE FOR THE PROFESSIONAL DEVELOPER

www.developercareers.com

www.developercareers.com

Imagine an IT job where you get world-class training and your department is never under-staffed.

Or go to dice.com and actually find one.

dice.com
High tech jobs online

110,000 high tech jobs, including your next one.

AN EARTHWEB SERVICE

Software Careers | THE ULTIMATE JOB RESOURCE FOR THE PROFESSIONAL DEVELOPER

Software Careers | THE ULTIMATE JOB RESOURCE FOR THE PROFESSIONAL DEVELOPER

Key Benefits

- Largest and most efficient reach of software professionals available through any recruitment media.
- Reach into the universe of developers with the most-sought-after skills in: Windows, Databases, Web, Java, C++, and multi-language skills.
- Make a high impact by gaining high frequency of exposure to readers in multiple magazines.

Elizabeth Doherty
650-655-4193 • edoherty@mfi.com

Call today and don't miss your chance to be a part of an indispensable career resource for developers!

WHERE THE HUNTER BECOMES THE HUNTED.

CareerShop.com
Career Enhancement For The Computer Generation

You're talented and capable, right? A desirable target in today's job market. So why should YOU have to beat the bushes to track down a position? Especially when we'll deliver potential employers right to your e-mail box – confidentially. Provide us with a little information, and we'll dangle your qualifications in front of thousands of prospect-hungry employers. We offer numerous career enhancement opportunities for computer-oriented professionals – making you even more attractive to top companies. So visit our Web site to get started. And remember – your career hunt ends at

WWW.CAREERSHOP.COM

FOR MORE INFORMATION, CALL 800-639-2060.

PUT YOUR FREE
PERSONAL JOB SHOPPER
TO WORK TODAY!

SEE OUR WEB SITE FOR DETAILS!

the effect of forcing the contents of register 1 into a state consistent with the value measured in register 2. This can still be a superposition of values, but the values all have to satisfy *x to the a mod n equals k*.

The algorithm then computes the discrete Fourier transform on register 1. Again, this is a one-step operation, thanks to the magic of quantum parallelism. This has the effect of placing register 1 in a state where observing its value gives a number that has a high probability of being a tractable function of *r*, the desired period of the function. Anyway, that's what they tell me.

A conventional computer then uses *r* to finish the factorization. *Please* don't build a quantum computer and try to run Shor's algorithm on it based on my description; I undoubtedly got something wrong. I just wanted to give you the flavor of it, which is all I was able to extract from the primary sources.

Meanwhile, though, research into actually building a quantum computer, not just a simulation, slogs along at various research sites throughout the world.

Some labs have actually created working qubits, which is no mean accomplishment. Yasunobu Nakamura and his colleagues at NEC Fundamental Research Labs in Tsukuba, Japan, have come the farthest to date in constructing a device capable of remaining in the necessary quantum state long enough to do some computation (all of two nanoseconds). Their device uses a superconducting quantum dot held at a temperature of 0.03K, so don't look for the home hobbyist kit version any time soon. The next step in the gargantuan bottom-up process of building a quantum computer is to figure out how to wire up a few qubits to make a logic gate. And as the Raw microchip project shows, you can do a lot with logic gates.

Sun's Latest Trick

Sun has released details of its MAJC (pronounced "magic," which stands for "Multiprocessor Architecture for Java Computing") chip architecture. As I understand it, MAJC was developed to move and manipulate nontext data efficiently and to promote Java, although that's not exactly how they put it in the presentations, which describe such MAJCal features as:

- Multiple levels of parallelism. There's parallelism in the handling of data through SIMD instructions, at the instruction level via multiple functional units per processor, at the thread-of-execution level via support for multithreading, and at the system level because you can have multiple MP processors on a single chip.
- Software portability. MAJC chips are designed for installation-time or JIT compilation of code. Compatibility across generations is regarded as somebody else's responsibility; it won't be a constraint on the design of future chips.
- Modular architecture. Because Sun sees MAJC chips being used in everything from screen phones to medical imaging devices, the technology employs a modular architecture: One implementation might be just one processing unit and two functional units with unneeded instructions turned off, while another might involve many processing units, each with four functional units. The desired design would be put together from modular components.
- Other stuff. It's based on a Very Long Instruction Word approach. More news about it will have been released by the time you read this, but this much seems clear: while this may be a very useful device, it will not in and of itself seem very magical. The applications of it might.

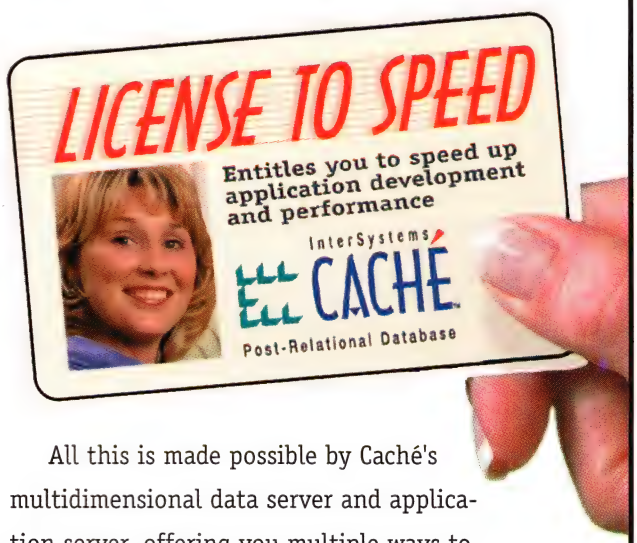
DDJ

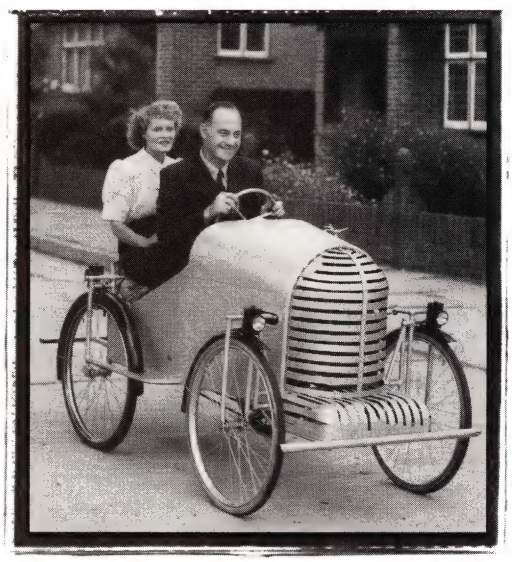
With a Caché database license, you can speed up your development.

With Caché, the post-relational database, you can develop complex applications so fast it's like owning a "License to Speed."

With advanced object and Web technologies, Caché can dramatically speed up your time-to-market. But Caché gives you more than a rapid application development environment.

Once deployed, Caché-based applications are so fast we've had customers get up to a 20x boost in SQL performance after switching their database. And without changing a single line of application code.





Do you still peddle applications that run on a relational database engine?

All this is made possible by Caché's multidimensional data server and application server, offering you multiple ways to access data, multidimensional storage architecture, scalable performance, plus richer technology for faster Web and object development.

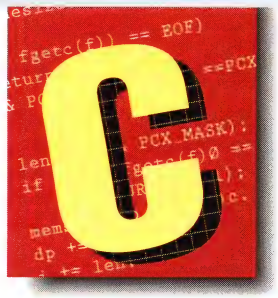
Caché comes from InterSystems, the leader in high-performance database systems, with over 2,000,000 licensed users worldwide, 20 years of database experience, and 24 x 365 support.

If you want to speed things up, start by moving quickly to call us at **1-800-753-2571.** We'll rush information to you about the DBMS that is so advanced, it won the "Best New Database of 1998" award.

InterSystems
CACHÉ
Post-Relational Database

www.LicenseToSpeed.com One Memorial Drive, Cambridge, Massachusetts 02142 1.800.753.2571

© 1999 InterSystems Corporation. All rights reserved. Caché and "License To Speed" are registered trademarks of InterSystems Corporation.



Teaching C++ for the Sixth Time

Al Stevens

It's August in the mountains of central eastern Pennsylvania. We're here in the RV for Judy's annual family reunion, to be hosted this year at the farm where she grew up. The drought ended last week with some heavy rain, and everything looks green again, although the crops that my father-in-law planted this spring suffered the effects of too much heat and too little water. There's not much to pick.

The farm sits on the southern side of a small mountain range overlooking a wide, green valley. The tiny town of Brandonville nestles at the base of the valley. During the winter when the trees are bare, you can see a few rooftops down there. This time of year only a distant neighbor's farmhouse reminds you that there is anyone else in the world at all. If you are accustomed to the sounds of a city or live anywhere near an interstate highway you would be stunned by the silence here. My father-in-law is 80, has been a farmer all his life, and has owned and worked these 40 acres for the better part of 50 years. His comment when I marvel over the spectacular view is, "If it wouldn't be for those damn trees, maybe we could see some more of the town."

This trip brings to a conclusion a two-week visit to our house in Florida by our four grandchildren, ages five, seven, seven, and nine. They returned with us to rejoin their parents who look rested and content. You don't want to know how I look. I'll spare you the details of what it's like to keep four little people entertained, fed, clean, well-behaved, out of trouble, and from killing one another from morning to night when you are not used to doing it and they know it. I will say this much: If you want to know anything at all about the Cartoon Network, Power Rangers, or Pokémon, just ask me.

Al is a DDJ *contributing editor. He can be contacted at astevens@ddj.com.*

That hectic visit was not the only harried experience brought to a close this week. I just completed work on the sixth edition of a C++ tutorial book called *Teach Yourself C++*, published by MIS:Press, which is now owned by IDG Books, who publish all those *For Dummies*... books. (There is a *C++ for Dummies*, written by someone else, but I've never read it. I don't believe that dummies can do C++.) The first five editions of my book reflect the evolution of the C++ programming language as it has been implemented on PC platforms and as the ANSI/ISO standardization effort has proceeded. I didn't plan it that way. The language kept growing and changing, the compilers kept upgrading, and I kept revising the book. The sixth edition will be the first one to be published since the Standard was approved.

The sixth edition departs from the traditional book-with-code format that I used in the first five editions. Some publisher-type suits in positions of influence took some of the money they made by selling books to all those dummies, acquired MIS:Press, took a look at its inventory of titles, and decided that what had been a best-seller needed to be fixed and made more "visually appealing." Publishers, you see, know much better than authors and readers what constitutes a good programming book and what appeals to programmers. I don't know how they know that, but they do. Just ask one of them.

The new layout uses what they call a "spread" format. A spread comprises two facing pages with narrative text and snappy "visual elements" like hints, references, and notes on the left page and more visual stuff—figures, screen shots, and so on—on the right. Each such spread teaches what they call a "task" or "concept." They developed this format for books that teach things that have many visual elements, such as applications, HTML, and Visual Basic, and they think that it should work for advanced programming language books, too. Programming books for the MTV generation, the sound bite set, the *People* magazine crowd. Short sentences, lots of eye candy. Too many words and not enough pictures scare those dummy book buyers off. Before long, computer books will include a companion package of crayons instead of a CD-ROM.

Visual appeal, that's what sells. But with a generic C++ book, the only thing visual is source code and program output. Discreet lessons about C++ often involve more code than you can get on a page, or more text, or too little code to have its own page, or too little text, or some combination of all that. Rarely does a specific lesson involve exactly enough text and code to fit this new, visually appealing, Johnny Bravo, dumbed-down format. Try, for example, to fill two pages explaining and demonstrating what an *int* is and how to instantiate one. Try to explain and demonstrate "explicit specialization of a member of a class template partial specialization" in only one page of text and 30 lines of code. Try explaining to a development editor that there are no pretty CROSS REFERENCE visual elements with this concept because THERE IS NOTHING TO CROSS REFERENCE. Duh. Consequently, an author spends a great deal of time contriving content in some cases and sacrificing content in others to force the problem to fit the solution. Writing this book was not an easy experience. I'll take rowdy grandchildren every time.

Don't feel too sorry for me, though. I whined a lot throughout the project, and they gave me more latitude than I think they would have liked. Mine was the first programming book in the new format and we were learning about it together.

It was my hope that the sixth edition of *Teach Yourself C++* would be the last one, particularly given the pain that writing it involved. The C++ Standard is approved and

the language is well-defined and understood. However, the last several editions of the book included compilers on the companion diskette and, later, the companion CD-ROM, and I am not willing to sacrifice that feature. As we speak, I do not have a standard conforming C++ compiler and library that I can freely distribute. The fourth edition included the DOS DJGPP port of the GNU command-line compiler. The fifth edition included the Cygnus gnu-win32 port of the GNU C++ compiler along with a front-end Windows 95-hosted IDE named Quincy 97 that I developed for this purpose. The sixth edition will include an improved Quincy 99 with the mingw32 port of the Egcs 1.1.2 compiler, which is a closely conforming C++ compiler. Its library, however, is still the old one, so my book, which purports to be about Standard C++, is unable to demonstrate some nifty features of the Standard C++ Library, such as stringstreams, wide-character streams, and so on. This means that whenever they get all that stuff working, I'll be looking at doing a seventh edition. Oh, the pain.

Frameworks, Graphics, and C++

Although my new edition promises to be a hot seller (according to the suits), most of you will not need to read it. Most readers of this column already know C++. Many of you have told me that you learned C++ from earlier editions of the book. (You know, the ones in the old format that wiser heads assure me no one would ever buy.) But during its development, I learned some new and better ways to teach some difficult subjects. Difficult because arcane concepts need concrete examples.

One of the things that the mingw32 port of Egcs adds to the compiler is support for Windows applications development with the Win32 API. Ho hum. But wait. This feature opened a window of opportunity that I did not have with earlier editions — the ability to demonstrate many of the features of C++ and object-oriented programming by using simple graphical applications.

In the old days, if you wanted to write about graphics on a PC platform, you had to spend some time talking about VGAs, XVGAs, and the like. Open any graphics programming book from that time and you'll find lots of details that you don't want to know about. The authors had no choice. If they were going to get their example programs running and displaying graphical images, they had to deal with the hardware. GUI platforms obviate all that low-level stuff by putting a device-independent layer between the programmer and the hardware.

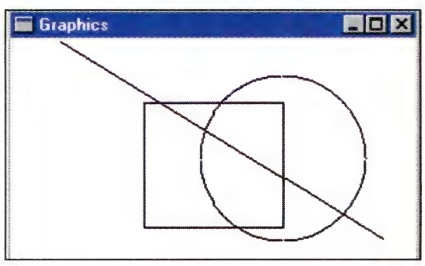


Figure 1: *Using the TYFC framework.*

A C++ tutorial book should teach C++, not some GUI API. But there is a compelling reason to use a GUI to teach C++. Nothing demonstrates the properties of object-oriented programming tools better than a simple graphical class library and a GUI simplifies the graphical parts. Every programmer who has ever written or used graphical rendering programs that draw shapes on the screen can immediately understand polymorphism when you explain it in terms of abstract and concrete shape classes. The abstract base shape class knows nothing about specific graphical shapes other than the properties that they all share — position, rotation, and so on. Concrete shape classes derived from the abstract shape know how to draw, scale, transform, and rotate themselves. Early works on OOP missed the boat on this one by trying to use examples about furniture, insects, toasters, and such to explain class hierarchies. These are not things that you typically deal with. Graphical objects are, and they are visual things that are easily demonstrated with concrete examples. More "visually appealing," don't you know.

There is another reason to use a GUI to teach C++. You don't want to burden the student with details of the API, which is a course in itself, yet the example programs need the API to display the windows. To hide those details, you can provide a simple framework. Most C++ programmers use GUI frameworks such as MFC. By offering early exposure to a simple framework, you are preparing the student for the real world.

TYFC

To teach all these things, I built a simple framework called TYFC, for "Teach Yourself Framework Classes." Catchy, huh? The version I discuss here is more advanced than the one in the book. I have experimented with TYFC to further examine its potential as a teaching aid, and this discussion reflects that research.

TYFC implements an abstract base class named *TYFCApp*, from which the application program derives a class. *TYFCApp* is small, consisting of one header file, tyfc.h (available electronically; see "Resource Center," page 5), which declares the *TYFCApp* class, and one source code file, tyfc.cpp (available electronically), which defines the member functions. You do not need to understand the implementation details of *TYFCApp*, which teaches the lesson that hiding details of implementation raises the programmer's level of abstraction.

You are allowed to peek, however. Another lesson to learn about C++ is that although the details are said to be hidden, they are actually mostly in view in the class declaration and completely in view in templates. You can hide nontemplate member function definitions, but I prefer that the source code be available, if only for documentation.

The application program derives a class from *TYFCApp*. The derived class represents the application, and its constructor specifies to the base class constructor the size of the application window and its caption. The derived class overrides the base class's pure virtual *Initialize* function. The program instantiates an external object of the derived class. As a result of that instantiation, *TYFCApp* creates the window and calls the derived class's *Initialize* function, which must put together something to display. textgl.cpp (available electronically) demonstrates the application program's use of the framework. Observe that the application has no *main* function. The framework provides it, in this case, a *WinMain* function, a detail that TYFC hides from you, another lesson to be learned. The *WinMain* function ensures that there is an instantiated derived *TYFCApp* object somewhere (its constructor records that fact, which is why the application object must be external), creates a window, and enters the usual message dispatching loop. The application program's *Initialize* function instantiates some graphical objects and calls *addshape* to add them to the display, in Figure 1.

TYFCApp, being a teaching vehicle, is only for static graphical displays, although it could evolve into a much larger framework. It knows about an abstract base class named *Shape*, and it maintains an *std::list<>* container of pointers to *Shape* objects, teaching the student that standard containers are easier to use than homegrown ones. *TYFCApp* also demonstrates polymorphism. Whenever the run-time system sends the WM_PAINT message to the application window, *TYFCApp* calls the *draw* function for each of the *Shape* objects pointed to by entries in the list container. The *draw* function in the *Shape* class is a pure virtual function, so the pointers have to be pointing at objects of classes derived from *Shape*, and those classes must have overriding versions of the *draw* function.

I used an *std::list<>* container for the *Shape* pointers, because a future version

Another product released too soon!

Don't Let Product Defects Take Your Business Down In Flames!

You need the very best defect tracking solution to keep your business soaring. Only at DevTrack.com will you find a list of every major defect tracking product on the market. Take a look, and decide for yourself which one is best for you. Once you've seen the others, you will choose DevTrack.

Check out our competitors today @ www.devtrack.com!

Try out the complete DevTrack solution for yourself – FREE!

www.devtrack.com

DevTrack

TechExcel, Inc. (800) 439-7782

(continued from page 108)
of the framework might need to know about and maintain the Z order of graphical objects. A list container is a good vehicle for reordering the objects it contains. Another lesson taught.

All *TYFCApp* knows about graphics is how to list *Shape* pointers and call their *draw* functions and how to draw a single point at an x/y coordinate on the screen. It adds shapes to its list when the derived class's *Initialize* function calls the *addshape* function. The application must use a graphical class library to exploit these characteristics. Win32 supports higher graphical abstraction, but *TYFCApp*'s purpose is to teach at lower levels. All we want the graphical engine to do is open a window let us draw pixels.

graphics.h derives classes from *Shape*, and it has several lessons to teach. First is the example of concrete classes defining the polymorphic behavior of their common abstract base class, which is the original reason for these libraries. But other lessons fell out of the exercise.

The first such lesson is about a different kind of abstraction, the kind you implement with templates. Templates aren't just for containers, they are also good at expressing abstractions. In this case, the graphical library in graphics.h does not need to associate with the specific graphical engine that *TYFCApp* implements. That detail is provided as a parameterized type, which makes graphics.h and its concrete shape classes portable to other graphical engines. The engine and the library share only the knowledge of the abstract base *Shape* class (available electronically), which is independent of either of them. To port graphics.h to a different graphical engine, you could put a wrapper around the engine to contain and manage shape pointers, provide the *addshape* and *drawpoint* functions, call the application's *Initialize* function, and call the shape objects' *draw* functions.

The next lesson is about specialization. Given that some graphical engines support higher level drawing functions, you can specialize the *draw* member functions for instantiations with those engines. Graphics.h implements the Bresenham line-drawing algorithm and the usual circle-drawing algorithm that plots points around the circle by using radians, sines, and cosines. These algorithms are nice for teaching how graphical objects are drawn, but they are slow and usually unnecessary. You can specialize the draw member functions for the *Line* and *Circle* template classes when they are instantiated for graphical engines that draw these things for you.

The last lesson from graphics.h is about namespaces. Graphics.h defines a *Line* class, a *Rectangle* class, and a *Circle* class. These are nice, meaningful names—the kind of names you like to give to things in a program. There is a catch, however. Somewhere in the mingw32-egcs 1.1.2 implementation of the Win32 API global namespace is something named *Rectangle*. Graphics.h puts its classes in a namespace to isolate its names from the global namespace and, consequently, demonstrates the utility of the C++ namespace feature.

The further I get into this specific teaching paradigm, the more I learn about how a simple graphics library can teach C++. Eventually the library will become too complex to serve that purpose, whereupon it might become a mechanism for teaching graphics programming in C++. And thus what started out to be a simple example of a complex subject surprises us when it demonstrates through its own growth how we learn things and how we ought to teach things. Sometimes you just can't see the town for the trees.

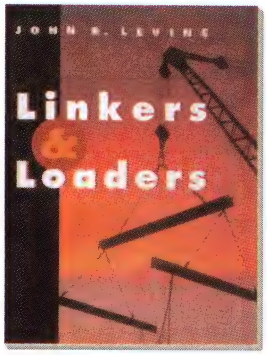
DDJ

Chart FX 98 is available only in a 32-bit version. Chart FX 3.0 is still available for 16-bit applications. Chart FX 3.0 and Chart FX 98 are royalty-free for distribution within your applications. For more information, call 800.392.4278. ©1998-1999 Software FX. All rights reserved. Visual Basic and COM technology are trademarks of Microsoft Corp.

MORGAN KAUFMANN PUBLISHERS

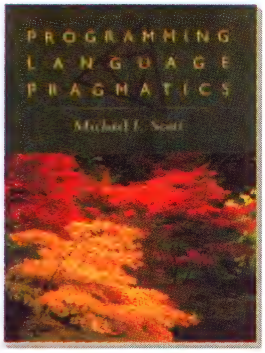
We Have The Solution

www.mkp.com



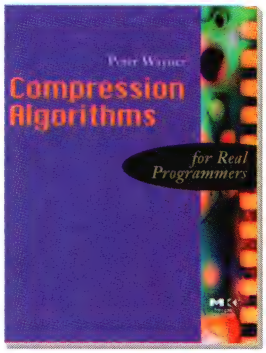
LINKERS & LOADERS
John R. Levine
1-55860-496-0; 300 pages; $36.95

Whatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. But do you know how to use them to their greatest possible advantage? Only now, with the publication of *Linkers & Loaders*, is there an authoritative book devoted entirely to these deep-seated compile-time and run-time processes.



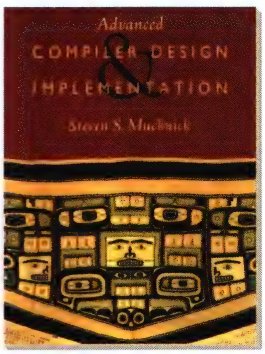
PROGRAMMING LANGUAGE PRAGMATICS
Michael L. Scott
1-55860-442-1; 904 pages; $69.95

This book addresses the fundamental principles at work in the most important contemporary languages, highlights the critical relationship between language design and language implementation, and devotes special attention to issues of importance to the expert programmer.



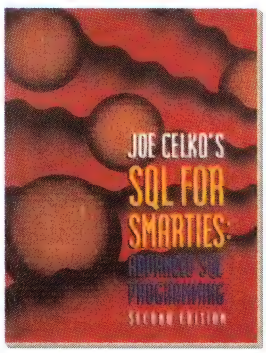
COMPRESSION ALGORITHMS FOR REAL PROGRAMMERS
Peter Wayner
0-12-788774-1; 350 pages; $44.95

This book offers practical, focused advice on using these standards to meet specific programming needs. The author examines all of today's leading compression techniques, emphasizing strengths and limitations while explaining how each works and the real-world challenges itís most suited to meet.



ADVANCED COMPILER DESIGN IMPLEMENTATION
Steven S. Muchnick
1-55860320-4; 1100 pages; $89.95

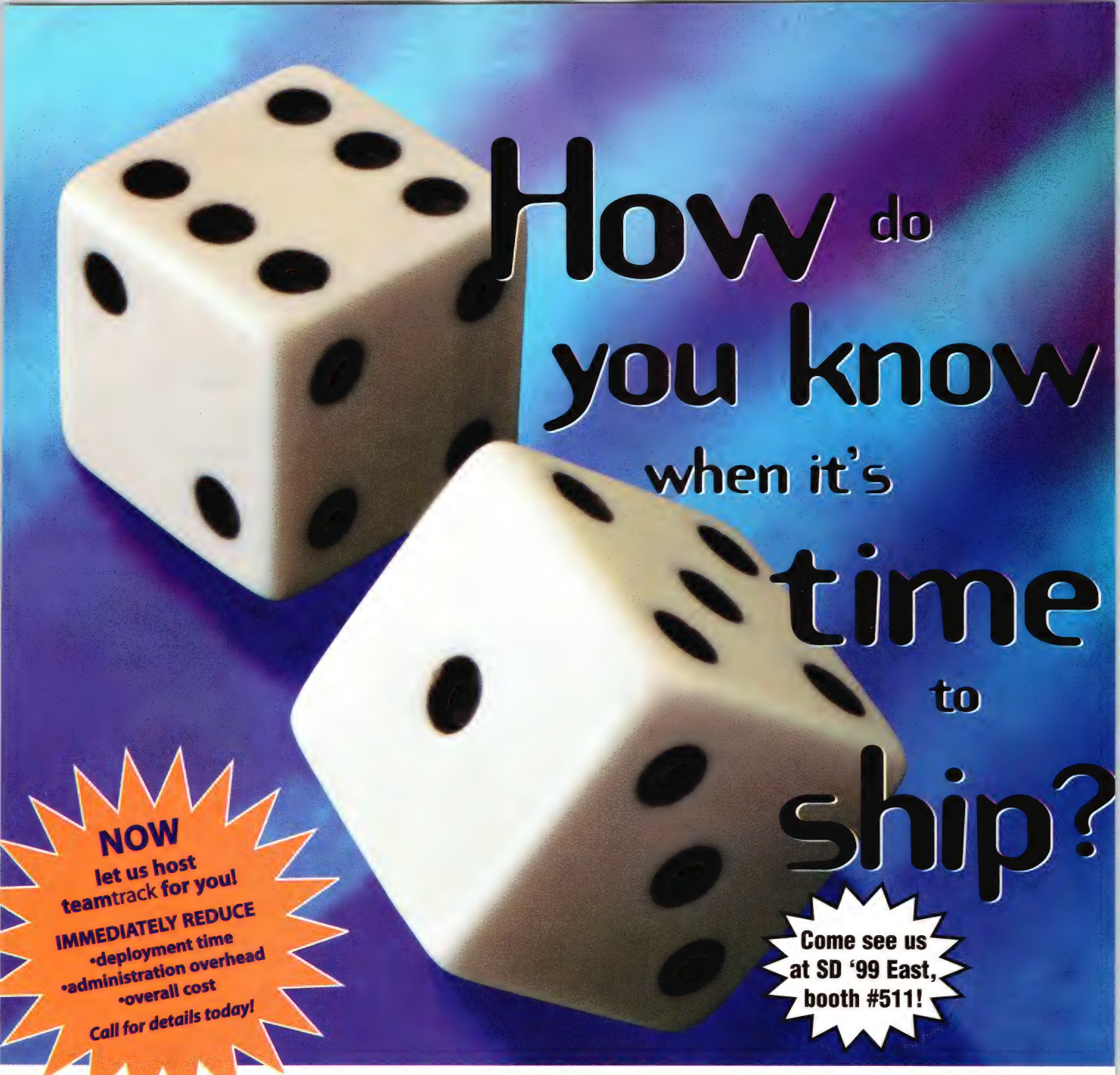
Comprehensive and up-to-date, this work provides computer professionals and graduate students with an understanding of advanced issues in the design and implementation of compilers for modern processors. This book helps the reader design efficient structures for highly optimizing compilers for real-world languages, determine the relative importance of particular optimizations, and select the most effective and efficient methods to use in implementing them.



JOE CELKO'S SQL FOR SMARTIES: ADVANCED SQL PROGRAMMING, SECOND EDITION
Joe Celko
1-55860-576-2; 448 pages; $44.95

SQL for Smarties was hailed as the first book devoted explicitly to the advanced techniques you need to transform yourself into an expert SQL programmer. Now, in this fully updated second edition, SQL mastermind Joe Celko keeps you moving forward, using his entertaining, conversational style to teach you the best solutions to old and new challenges and to convey *the way you need to think* if you really want to get the most out of your SQL programming efforts.

Phone: 800.745.7323/407.345.3800 Fax: 800.874.6418/407.345.4060 Email: orders@mkp.com Look for our titles at your local book store!



FINALLY, a problem-tracking system that takes the gamble out of software releases.

When 40% of development projects run over budget or beyond deadline and another 30% get cancelled altogether, what are the odds for your release?

Take control of the issues that threaten product quality and ship dates. **team**track has the power, flexibility and scalability essential for effective problem tracking. Plus more. Because only **team**track is built on a full Web architecture that provides all system functionality through ordinary Web browsers. So you get power without administrative pain. A customizable workflow engine, definable fields and forms, robust reporting, and integration with email and version-control systems are just the start of it.

Right out of the box. Right over the Web. It's a snap.

And suddenly, your software release is a sure bet.

Download free software at http://www.teamshare.com

teamshare™

teamshare, inc. • colorado springs, colorado • info@teamshare.com • 888-teamshare

teamtrack® 3

It works the way you want it to.

Does Java Support Design by Contract?

Krishnan Rangaraajan

Design by Contract (DBC), pioneered by Bertrand Meyer (see *Object-Oriented Software Construction*, Second Edition, Prentice Hall, 1997), is widely acknowledged to be a powerful technique for writing reliable software. The three key elements of DBC are preconditions, postconditions, and class invariants. An example of a language that has direct support for DBC is Eiffel. Unfortunately, Java does not directly support DBC. However, there are third-party tools for Java that extend support to Design by Contract. One such tool is Reto Kramer's freely available iContract (http://www.reliable-systems.com/tools/). iContract parses Javasource code that is annotated with assertions inside Javadoc-style comments and generates modified Java source code where the assertion checking logic is appropriately embedded. The benefits of this approach are:

- Because assertions are embedded inside comments, they are transparent to the Java compiler.
- No costly source modifications are needed to regenerate assertion-disabled byte code; simply recompiling the original source with a normal Java compiler is all that is needed.

Two other equally interesting approaches for incorporating DBC in Java are Handshake (see "Adding Contracts to Java with Handshake," by Andrew Duncan and Urs Holzle, Technical Report TRCS98-32, Department of Computer Science, University of California, Santa Barbara, http://www.cs.ucsb.edu/TRs/ TRCS98-32.html) and jContractor (see "jContractor: A Reflective Java Library to Support Design by Contract," by Murat Karaorman, Urs Holzle, and John Bruno, Technical Report TRCS98-31, Department of Computer Science, University of California, Santa Barbara, http://www.cs.ucsb.edu/TRs/TRCS98-31.html). Handshake uses a separate contract file to dynamically synthesize a class file that incorporates assertions, whereas jContractor uses a design pattern approach. Each approach has strengths and limitations, and it is not my intention to discuss them. In this article, I'll present yet another approach, one that is part of a class-testing environment available from Man Machine Systems.

In a nutshell, our approach is this: We parse a Java source file suitably annotated with preconditions, postconditions, and invariants inside Javadoc comments and generate triggers in JMScript (a proprietary scripting language that is built on top of Java). These triggers are then executed by running the Java program (along with the triggers) by attaching the script interpreter to JVM via an extension DLL.

The first step is to add preconditions, postconditions, and invariants to Java code within Javadoc comments. Special tags are used for this; Table 1 lists these tags.

- Class Invariant (@inv) should be defined inside a Javadoc comment before the class definition. The marker is followed by a Boolean expression that may reference any element of the class or its direct/indirect bases, including private elements.
- Precondition (@pre) must be defined within Javadoc comments preceding the respective methods. The marker is followed by a Boolean expression, as in the class invariant. Also, the condition may reference arguments passed to the method.
- Postcondition (@post) is specified within Javadoc comments preceding the respective methods. The condition may be any Boolean expression, as in the precondition case. The Boolean expression may also use the qualifier *$prev* on an expression to denote the expression's value at the method's entry point. For example, *$prev (top)* denotes the value of variable *top* at method entry (without the qualifier, it indicates its current value). The keyword *$ret* may appear in the condition to denote the method's return value.

The assertions must conform to the syntax of JMScript Boolean expressions. To capture more complex conditions, you can use the special escape marker "@macro" to embed a JMScript code fragment. Table 2 shows how preconditions, postconditions, and invariants, are used in the context of a class.

We allow the contract to be specified for an interface, which gets propagated to its implementing classes. The same rule is applied to class hierarchies. For both interfaces and class hierarchies, preconditions are disjuncted, whereas postconditions and class invariants are conjuncted.

Consider Listing One, the canonical stack implementation in Java (to avoid confusion with JDK *Stack* class, I've named the example *MyStack*). Once a Java source file has been annotated as the one just mentioned, the next step is to use the preprocessor *jmsassert* utility to generate JMScript triggers for the assertions. In this case, we run *jmsassert -s MyStack.java*. The preprocessor creates the output files default_MyStack.jms and Startup.jms. The first file (Listing Two) contains JMScript triggers corresponding to the embedded assertions in Java source. The second file (Listing Three) makes it convenient to register the automatically generated triggers with JMScript at run time.

The third step is to compile the relevant Java source files. Assume that in addition to the *MyStack* definition as just mentioned, we also have the test driver class in Listing Four. This driver, along with the *MyStack* class, is compiled using a regular Java compiler such as "javac." The final step is, of course, to run the Java code with assertions enabled. To do this, invoke the Java interpreter using: *java-Xdebug -Xnoagent -Djava.compiler= NONE -Xrunjmsdll:Startup StackTest*. The CLASSPATH environment variable must be appropriately set to JDK1.2 run-time files, and additionally, must include mmsclasses.jar, which is supplied as part of the assertion environment.

The DLL jmsdll includes the JMScript interpreter. This DLL registers itself with JVM and assigns assertion triggers to the respective Java methods. When the JVM invokes a method, the call is intercepted by the DLL, and if a trigger (precondition, postcondition, or invariant) is associated with it, the corresponding JMScript trigger method is invoked. The trigger code in JMScript can

Krishnan is president of Man Machine Systems and can be contacted at mms@giasmd01.vsnl.net.in or http://www.mms-india.com/.

Any application.
Any platform.
The only solution.™

Deploying enterprise software to multiple platforms? Whether they're applications, libraries, or servlets written in Java or platform-specific code, InstallAnywhere is the only solution that creates a single, universal installer that can deploy software via the Internet or CD-ROM to all platforms, including Windows, Solaris, Linux, Mac OS and others. Companies such as Sun Microsystems, Adobe and FedEx rely on InstallAnywhere to deploy software across multiple platforms, scaling from the smallest applet to the largest client/server applications.

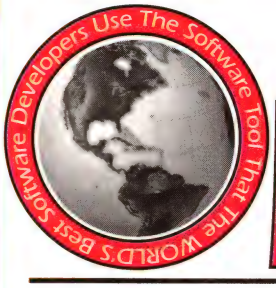
www.ZeroG.com

InstallAnywhere®
ENTERPRISE EDITION

zero
ZERO G SOFTWARE

© 1999 Zero G Software, Inc. InstallAnywhere, Zero G Software, the Zero G logo, and ZeroG.com are trademarks or registered trademarks of Zero G Software, Inc. All other trademarks are the property of their respective owners.

Get your demo fast www.opussoftware.com



OPUS MAKE

The compexity of software development in today's world demands a make utility that won't waste your time. **OpusMake is fast** (up to 4x's faster than other makes), **powerful** (our underlying technology relieves many performance hurdles) and **versatile** (an OpusMake makefile can be run on many different OS's)

To build your software quickly and reliably, only OpusMake makes it! *Because you don't want to run in circles getting your software to the world.*

Multiple O/S Support
MS-DOS / Win95
WinNT / AIX
SUNOS / Solaris
HPUX / Linux
And more...

Version Control Support
Intersolv PVCS
Microsoft SourceSafe
MKS Source Integrity
Burton System TLIB

Makefile Compatible
Intersolve PolyMake
Intersolv Configuration Builder
Microsoft NMake
UNIX makes

Lifetime Support
Free lifetime technical support

Money-back Guarantee
Get your money back for up to 60 days.

Opus Software, Inc.
1032 Irving St., Suite 439
SF, CA 94122
Tel: 800-248-6787
415-485-9703
Fax: 415-485-9704

©1998

Tag	Description
@inv	Class invariant
@pre	Method precondition
@post	Method postcondition

Table 1: *Tags to specify DBC elements.*

Event		Invariant	Pre-condition	Post-condition
Method Entry	Instance Method	Y	Y	–
	Constructor	N	Y	–
	Private Method	N	Y	–
	Static Method	N	Y	–
Method Exit	Instance Method	Y	–	Y
	Constructor	Y	–	Y
	Private Method	N	–	Y
	Static Method	N	–	Y
On Exception	Instance Method	Y	–	N
	Constructor	Y	–	N
	Private Method	N	–	N
	Static Method	N	–	N

Table 2: *JMScript Triggers Context.*

access private elements of a class. This is a significant advantage in terms of testing. Though JMScript is primarily useful for bringing DBC to Java, it's designed for general-purpose scripting, and has some attractive features such as partial function specialization, multimethod, dynamic inheritance, and so on. Its strength derives from the underlying JVM. (For more details on JMScript, go to http://www.mmsindia.com/.)

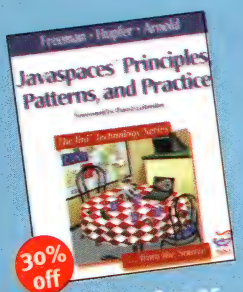
One of the benefits of this approach is that the original Java source code is unmodified; what is tested with assertions is the same as what is executed normally. To run the test driver without enabling assertion code, simply run: *java StackTest*. The limitation that Java source must be available in order to specify contracts is overcome when you use JVerify (a tool that supports invasive testing by executing test drivers written in JMScript). That environment allows assignment of triggers to Java by inspecting compiled class files.

Conclusion

Although Java does not yet directly support Design by Contract, there are several ways to add such support. The approach I've presented uses a preprocessor to map contracts embedded in Java source code to triggers in the scripting language. Triggers are then automatically executed by an extension DLL that includes the JMScript interpreter. To run without assertions, all that is required is to run the Java program without the extension DLL. This approach ensures that the released program is identical to the one tested. To test a class, no modifications are necessary (other than source annotation).

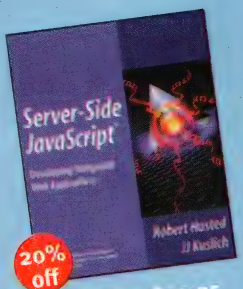
DDJ

(Listings begin on page 116)



30% off
List price: $39.95
Our price: $27.97
You save: $11.98

20% off
List Price: $19.95
Our Price: $15.96
You Save: $3.99

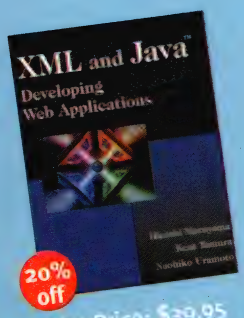


20% off
List Price: $39.95
Our Price: $31.96
You Save: $7.99

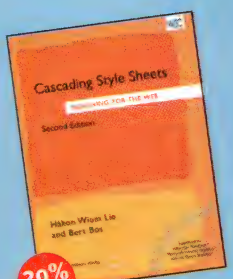
A Shopping List for E-Commerce Elves

Forget the wooden workbench. This holiday season, e-commerce elves know that networks must be synchronized, databases installed and tested, and a superior Internet interface ready to accept orders. Late night or last minute, Amazon.com has the books you need. With great discounts, reliable service and a bookstore just for developers, we'll make your holiday season hum.

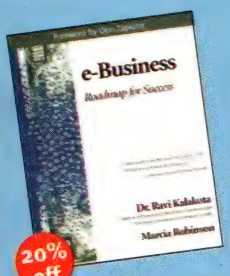
www.amazon.com/computers



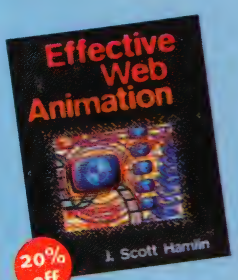
20% off
List Price: $39.95
Our Price: $31.96
You Save: $7.99



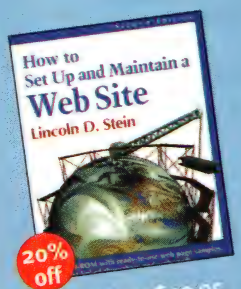
20% off
List Price: $39.95
Our Price: $31.96
You Save: $7.99



20% off
List Price: $39.95
Our Price: $31.96
You Save: $7.99



20% off
List Price: $44.95
Our Price: $35.96
You Save: $8.99



20% off
List Price: $39.95
Our Price: $31.96
You Save: $7.99

amazon.com
Earth's Biggest Selection.
books | video | dvd | music | auctions | e-cards | toys | electroni

© 1999 Amazon.com, Inc. All rights reserved. Amazon.com is a registered trademark of Amazon.com Inc.

Listing One

```
/**
    @inv (top >= 0 && top < max)
*/
class MyStack {
        private Object[] elems;
        private int top, max;
    /**
        @pre (sz > 0)
        @post (max == sz && elems != null)
    */
    public MyStack(int sz) {
            max = sz;
            elems = new Object[sz];
    }
    /**
        @pre !isFull()
            @post (top == $prev (top) + 1) && elems[top-1] == obj
    */
    public void push(Object obj)  {
            elems[top++] = obj;
    }
    /**
        @pre !isEmpty()
            @post (top == $prev (top) - 1) && $ret == elems[top]
    */
    public Object pop()  {
            return elems[--top];
    }
    /**
        @post ($ret == (top == max))
    */
    public boolean isFull() {
            return top == max;
    }
    /**
        @post ($ret == (top == 0))
    */
    public boolean isEmpty() {
            return top == 0;
    }
} // End MyStack
```

Listing Two

```
/*  Trigger file for class #default.MyStack. Generated by JMSAssert on
 * Monday, April 12, 1999. Any changes you make to this file will be
 * overwritten if you regenerate this file.
 */
import macro;

// Postcondition for method - MyStack(int)
MyStackPost(meth, $obj, sz) {
    assertPost(($obj.max == sz && $obj.elems != null));
}
// Precondition for method - MyStack(int)
MyStackPre(meth, $obj, sz) {
    assertPre((sz > 0));
}
// Postcondition for method - void push(Object)
pushPost(meth, $obj, obj, $ret) {
    assertPost(($obj.top ==
            this.top$prev + 1) && $obj.elems[this.top$prev] == obj);
}
// Precondition for method - void push(Object)
pushPre(meth, $obj, obj) {
    this.top$prev = $obj.top;
    assertPre(!$obj.isFull());
}
// Postcondition for method - Object pop()
popPost(meth, $obj, $ret) {
    assertPost(($obj.top == this.top$prev - 1) && $ret ==
                                        $obj.elems[$obj.top]);
}
// Precondition for method - Object pop()
popPre(meth, $obj) {
    this.top$prev = $obj.top;
    assertPre(!$obj.isEmpty());
}
// Postcondition for method - boolean isFull()
isFullPost(meth, $obj, $ret) {
    assertPost(($ret == ($obj.top == $obj.max)));
}
// Postcondition for method - boolean isEmpty()
isEmptyPost(meth, $obj, $ret) {
    assertPost(($ret == ($obj.top == 0)));
}
MyStackinv(meth, $obj) {
    assertInv(($obj.top >= 0 && $obj.top <= $obj.max));
}
static {
    assertStrMyStack = {
        { "<init>(I)V", "POSTCONDITION", "MyStackPost" },
        { "<init>(I)V", "PRECONDITION", "MyStackPre" },
        { "push(Ljava/lang/Object;)V", "POSTCONDITION", "pushPost" },
        { "push(Ljava/lang/Object;)V", "PRECONDITION", "pushPre" },
        { "pop()Ljava/lang/Object;", "POSTCONDITION", "popPost" },
        { "pop()Ljava/lang/Object;", "PRECONDITION", "popPre" },
        { "isFull()Z", "POSTCONDITION", "isFullPost" },
        { "isEmpty()Z", "POSTCONDITION", "isEmptyPost" },
        { "", "INVARIANT", "MyStackinv" }
    };
    setClassTrigger("MyStack",     assertStrMyStack);
}
load() {}
```

Listing Three

```
//Startup trigger file - c:\ranga\jverify\Startup.jms
import macro;
load() {}
static {
   default_MyStack.load();
}
```

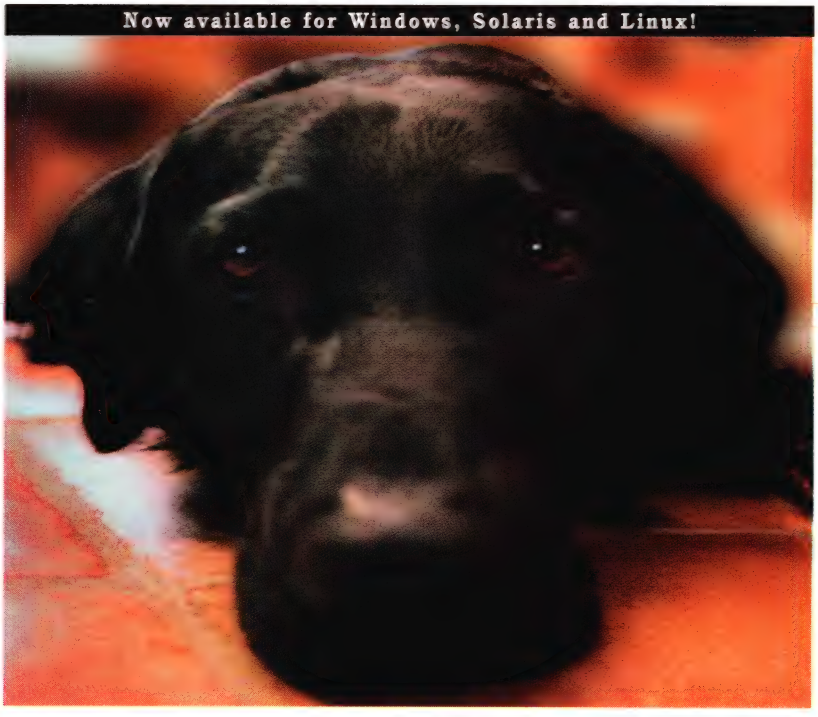
Listing Four

```
class StackTest {
    public static void main(String[] args) {
        MyStack s = new MyStack(2); // Can push at most two elements
            s.push(new Integer(1));
            s.push(new Integer(23));
            s.push(new Integer(0)); // Precondition violation here!
    }
}
```

DDJ

Put the bounce back in sluggish Java programs with OptimizeIt 3.0 Professional.

OptimizeIt's real-time data and easy-to-read graphs provide the information you need to solve performance bottlenecks, excessive object allocations, memory leaks. And with asynchronous data fetching, sampling and advanced profiling technologies, OptimizeIt lets you take on the largest Java environments.



What's more, OptimizeIt's CPU profiler lets you tune CPU time measurement to the most efficient level of detail for each test session — with low overhead. And our unique memory leak debugger actually highlights the references causing memory leaks — automatically.

So find Java problems before they become customer problems. Download your free trial at www.optimizeit.com today.

Optimize It!
www.optimizeit.com

Baaad Code!
Watch Java applications perk up when OptimizeIt™! does the training.

© 1999 Intuitive Systems, Inc. All rights reserved. OptimizeIt! is a registered trademark of Intuitive Systems, Inc. All registered trademarks belong to their respective manufacturers.

AUSTIN
BOSTON
LOS ANGELES
NEW YORK
CHICAGO
ATLANTA
WASHINGTON, D.C.
SEATTLE
MINNEAPOLIS
SAN FRANCISCO
Sandy Conroy, PROGRAMMER
(Now she designs web sites, too.)
Your web, your life, your conference.
WEB design & development
where web professionals converge
www.mfweb.com
PLATINUM SPONSORS
SILVER SPONSORS
IBM
Adobe
SCALA broadcast multimedia
interland We make the Web work for you.
real RealNetworks

Designed for

Microsoft®
BackOffice®

IBM, DB2 and DB2 Universal Database are trademarks of International Business Machines Corporation in the United States and/or other countries. Microsoft, BackOffice, the BackOffice logo, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both. ©1999 IBM Corp. All rights reserved.

DB2

runs natively on Windows NT® and meets every "Designed for Microsoft BackOffice" spec.

DB2® Universal Database™ takes Microsoft Windows® to another level: up to clusters or down to Windows CE. Visit www.ibm.com/software/data for a free developer's edition.

IBM®

Adaptive Unsharp Masking

Tim Kientzle

Unsharp masking is a photographic technique that increases the sharpness of a photographic image. It's heavily used by astronomers for improving blurry telescope images. More recently, it's become a standard tool among graphic artists for enhancing blurry scanned images.

In the darkroom, you create what is called an "unsharp mask" from a large-format negative by exposing another piece of film using a spacer. The result is an underexposed, blurry positive. When you sandwich this with the original negative, some of the fuzziness is canceled out, resulting in a sharper image.

The digital version of this process is actually much simpler in many ways. To demonstrate the technique, pretend for a moment that you're an astronomer trying to photograph a distant star. Figure 1 is a cross-section of the image you would like to get—a sharp-edged light disc against a dark background. But there are many kinds of interference to contend with, so your actual result is more like Figure 2.

Just as in the photographic technique, the key is to take the actual image and subtract a blurred version of it from itself. Figure 3 shows the result in this particular case. This unsharp mask is very faint; in this particular case, the maximum value is only about a hundredth of the peak magnitude of the original. This is an important observation; for example, the "Unsharp Mask" script included with version 1 of GIMP (http://www.gimp.org/) builds an unsharp mask using 8-bit graphics operations, which severely limits the enhancement you can obtain.

The enhanced image is created by adding a multiple of the unsharp mask back to the original. In the case of Figure 4, I've multiplied the unsharp mask by 100 before adding it back to the original. The enhanced version has a much narrower peak than the original data. The sides of this peak are much steeper, which corresponds to a sharper image.

You should note a few interesting facts about Figure 4. First, I created the unsharp mask using a blur radius much smaller than the blurring in the original; you can play with the parameters, but I think you'll agree that a relatively small radius gives the best improvement. Also, notice how the enhanced data here goes out of range (the red line goes below the x-axis). This is a problem with too-aggressive unsharp masking. You lose all detail in areas where this happens. One way to address this problem is to reduce the weighting when you combine the unsharp mask with the actual data. Of course, real images don't have a single feature you're trying to enhance, so it may be very difficult to find a single weight that simultaneously provides the needed enhancement and does not destroy faint details.

Adaptive Unsharp Masking

The key to creating an adaptive unsharp-masking algorithm is that all of the parameters apply to single pixels. It's therefore quite reasonable to vary these parameters from pixel to pixel to provide different levels of enhancement. To do this automatically, you first identify the problem areas. Figure 5 shows only the parts of the enhanced image that went out of range.

To build Figure 4, I added 100 times the unsharp mask at every point. In the adaptive version, I start with an array that contains 100 for every pixel. I then wanted to reduce the weight around problem areas. This is not an entirely trivial operation; I wanted to reduce the weight more in areas that were more troublesome. I also wanted the weighting to vary smoothly across the image so that my adaptive weighting wouldn't alter any details. (After all, the point of having an adaptive algorithm is to preserve details.) One simple solution was to first add a multiple of Figure 5, then apply a Gaussian Blur to the resulting array. Figure 5 shows more negative in areas that are more of a problem. By blurring the weighting array, I ensured

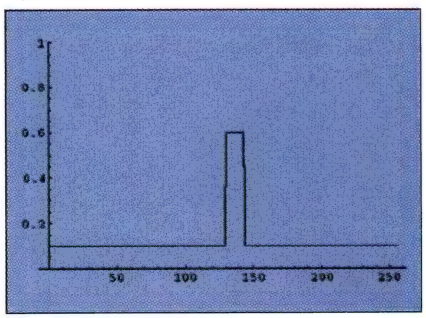


Figure 1: *The data we wanted; a light-gray block against a dark-gray background.*

Tim is DDJ*'s executive editor and can be contacted at kientzle@ddj.com.*

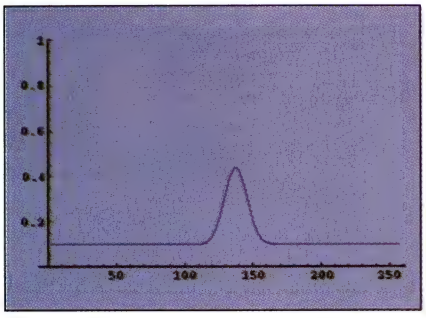


Figure 2: *A blurry version of Figure 1; note that the peak is much lower.*

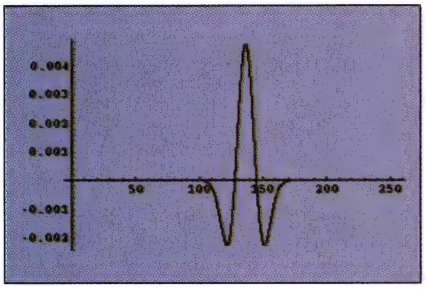


Figure 3: *The unsharp mask is the difference between the test data and a blurred version of that data. In this case, I used a Gaussian Blur with a radius of two, and subtracted the blurred version of Figure 2 from the original.*

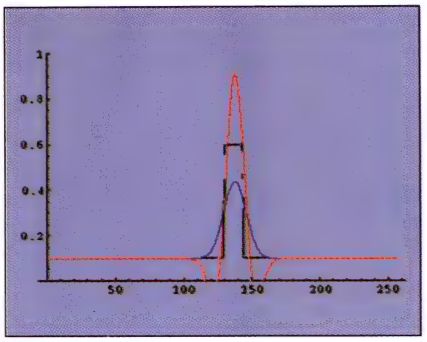


Figure 4: *Adding the unsharp mask to the test data yields an enhanced version (red), which I've shown together with the actual data (blue) and the desired data (black).*

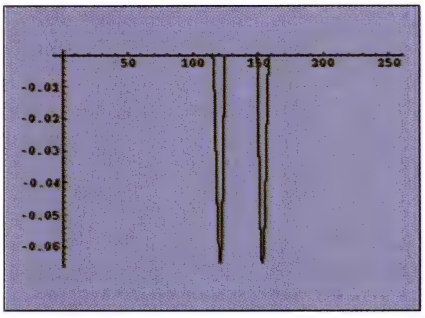


Figure 5: *The problem areas from Figure 4.*

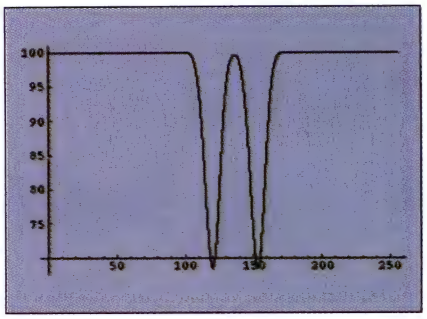


Figure 6: *I initialized my array of weights to 100, subtracted a multiple of Figure 5, and then applied a Gaussian Blur.*

that the weighting would vary smoothly across the image. The resulting array of weights is shown in Figure 6.

Once I had a suitable weighting array, I could multiply each pixel of the unsharp mask by a different weight and add the result to the original data. Figure 7 is the result. This final enhanced version does not go out of range.

The basic approach here was simply to replace a single fixed parameter (the weighting used to combine the unsharp mask with the original image) with a separate parameter for every pixel. Of course, this required juggling other parameters; I used a single

Configuration Management. Issue Tracking.

Razor®, a full featured solution at a realistic price. Razor gives you the edge in configuration management process implementation.

Lockheed Martin wanted a competitive advantage. They chose **Razor** as the preferred tool for file version control, release management, and issue tracking for the entire corporation.

NASA is rewriting the launch control software for the space shuttle. **Razor** is an integral part of implementing and enforcing their ISO 9001 compliant process.

Boeing, AT&T, Iridium, and Los Alamos National Labs are just a few of our leading edge technology clients that have similar success stories.

Need to know where you are going, where you have been, and how you got there? **Razor** will help you get the competitive edge—today!

TOWER CONCEPTS

Now Available for Windows NT and Linux!

(315) 363-8000 • www.tower.com

Copyright 1999 by Tower Concepts. All rights reserved. Razor is a registered trademark of Tower Concepts. All other trademarks mentioned are the property of their respective holders.
Tower Concepts is a subsidiary of Visible Systems Corporation.

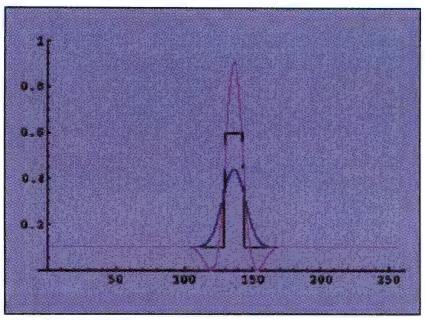


Figure 7: *The final enhanced data (purple). This was created by multiplying the weighting array (Figure 6) by the unsharp mask (Figure 3) and adding to the original data (blue). This enhanced version does not go out of range.*

multiplier and a blur radius to construct the weighting array. To choose that multiplier automatically, you can simply increase it just to the point that none of your original image is out of range. I don't know of a good way to automatically determine the blur radius for this operation.

The basic unsharp-mask algorithm does have one other fixed parameter that could become adaptive. A Gaussian Blur is a pointwise operation; each pixel is replaced with a weighted average of neighboring pixels, with the weighting determined by a Gaussian distribution with a selected radius. There is no technical reason the Gaussian radius cannot be varied at each pixel. To do so, you would need an array of Gaussian Blur radiuses that you could manipulate. The final blurred version would require, at every pixel, computing fresh Gaussian weights and applying those weights at that pixel. This is, unfortunately, extremely computation intensive, and probably not practical for most purposes. I've provided a Mathematica notebook (available electronically; see "Resource Center," page 5) that lets you experiment with the different parameters. The notebook uses a Fourier transform to compute the Gaussian Blur, which is a fast method that unfortunately does not lend itself well to an adaptive approach. (You can view the notebook using MathReader 3.0, which is available at no charge from Wolfram Research; see http://www.wolfram.com/.)

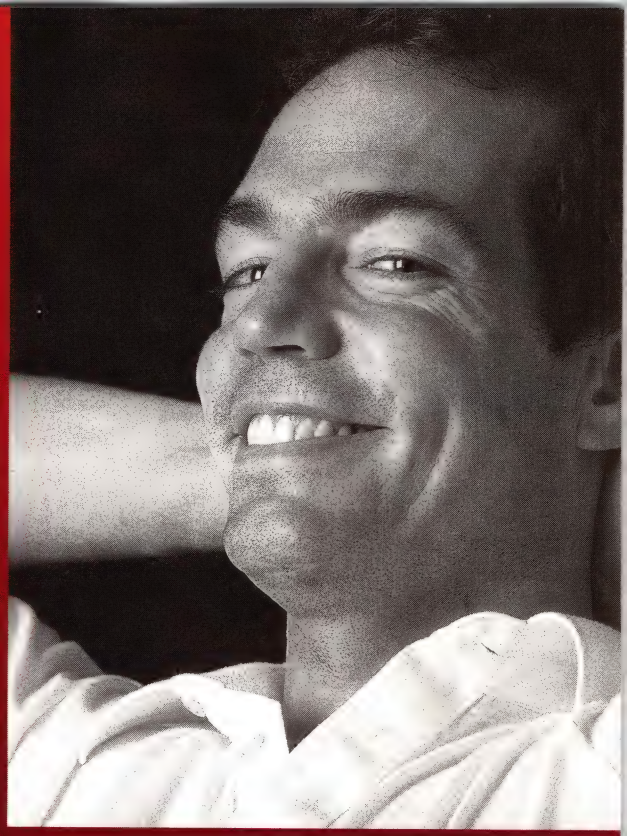
Conclusion

The unsharp-mask weighting layer doesn't necessarily have to be constructed automatically. An alternative approach would be to initialize it to all zeros, thus allowing users to paint on it. If carefully done, this would allow users to selectively sharpen the image only where needed.

DDJ

"My specialty is EDI. I wanted a more managerial role and more money. I announced my availability on dice so people coming to me would already know my qualifications. It was incredibly easy. You'll get calls very quickly. I had over a dozen in the first two days - and more interviews lined up than I knew what to do with. I'm very happy in my new job. In six months I'm going to click the dice again and go after even bigger money."

Gary B. *Dallas, TX*



I Doubled My Pay In Less Than A Week.

dice.com
High tech jobs online

AN EARTHWEB SERVICE

Calabaza

Dennis E. Shasha

"We have found a remarkable tribe in the jungles of Venezuela," the strongly built ethnobiologist Panta de Leon began in slightly accented English. De Leon had heard of Ecco from Natasha, the archaeologist of the Sudan. De Leon's motivation, we all decided later, was much more pure.

"The tribe has lived in near permanent isolation except for occasional, mostly unfriendly, exchanges with outsiders. They are fine singers and percussion musicians. Their principal musical instrument is a fruit gourd called (in Spanish) a Calabaza. This gourd is so central to their life in fact that they call themselves the Calabazas when meeting (friendly) outsiders. Recently, I discovered that they also use the Calabaza as a timepiece for the ritual they call—I'm translating roughly—'Prayers to the Clock God'.

"Let me explain. Their's is one of the wettest, most mosquito-infested corners of the world. This has kept them safe. It also means that anything made of metal either rusts or is quickly impregnated with water.

"Many years ago, they found a group of explorers who had lost their way and then died of a poisonous fruit. One of the explorers had a watch, which the Calabazas brought back to the village. The watch worked for one moon, they say.

"The first day they had it, the tribal elders tried to figure out the meaning of the timepiece and they quickly concluded that it must be used for prayer. The explorers died because they hadn't prayed at the right time, they decided. Fascinated by the circular rhythm of the second hand, the elders directed the village artisan to create several gourds of the same size, each with a hole. They told the artisan that he should create the gourds so that a full gourd would drain completely in one minute when the hole was opened. Unfortunately, the village artisan found himself unable to carve a large enough hole without destroying the Calabaza. Instead, he created Calabazas having two holes.

"Opening one hole in a full Calabaza would cause the water to drain completely in 11 minutes. Opening both holes would cause it to drain in 5 minutes. These were exact times, to the second in fact—a remarkable achievement considering that all his tools were handmade."

Liane perked up. "Why should two holes be more than twice as fast as one?" she asked.

"The vortex effect, I think," Ecco responded. "Two close holes will create a vortex and water will pass through both of them faster than the sum of the speeds of each hole alone."

De Leon nodded in agreement. "Whatever the reason, that was the effect. The village elders were unhappy with that, because they had already invented a full moon ritual in which percussion Calabazas would be hit every minute as the village women chanted.

"They told the artisan that he must figure out how to make a one-minute Calabaza by the next full moon. He told them that he could not, but that with five Calabazas, four assistants, a ready supply of water, and about an hour, he would be able to tell the drummers to hit their drums every minute exactly."

"How long does it take to fill an empty Calabaza?" Liane asked.

"Natasha told me you were clever," Panta said to the 11-year old. "Less than a second. One can just dip it into a big barrel. You can assume it takes no time."

"Well, then it shouldn't be so hard," Liane said. "For example, it's easy enough to measure six minutes. You start two Calabazas at the same time, opening one hole of one and two holes of the other. Six minutes is the time between the draining of the two-hole Calabaza and the draining of the one-hole Calabaza."

"I think you are on the right track," de Leon said. "But it is important that you figure this out today. You see, the full moon is about to come and the village artisan has taken terribly ill. They are counting on me to help them figure out what he did. They refuse to let me use my watch, as they revere the gourds as holy. Now for my personal appeal: As an ethnobiologist, I have made it my goal to win their confidence. They trust me so far and they have taught me about many wonderful herbs. They even have a remarkable medicine to enhance musical ability, seemingly without side effects, but I must stay with them longer before they will teach me about it. I would really like to help them with their problem first."

"Let me try a small example," Liane said. "Suppose that a one-hole Calabaza drained in three minutes and a two-hole Calabaza drained in two minutes (reversing the vortex effect for the sake of the example). Then it would be possible to measure minutes by allowing Calabaza A to drain in three minutes, B to drain in two minutes, and then repeatedly filling each Calabaza and draining it with two holes. So, A would mark times 3, 5, 7, 11, 13,...and B would mark times 2, 4, 6, 8, 10, 12,... Thus, it would be possible to mark minutes beginning two minutes after the procedure starts. Now let's try it for the gourds the artisan really builds."

Liane and Ecco talked for a while. Liane explained an idea. Ecco smiled.

"Mr. de Leon," Ecco said with a smile, "Liane has solved your puzzle. Five Calabazas seems to be the minimum possible. If five people follow this simple procedure, they will be able to mark single minutes starting 40 minutes after beginning the procedure."

Reader: Is five Calabazas really a minimum? Can you invent a procedure that involves filling Calabazas and watching them drain such that it is possible to mark single minutes starting 40 minutes into the procedure? Assume that it is easy to tell when a Calabaza drains. Can you do it better (either with fewer Calabazas or fewer minutes)?

After Panta de Leon left, Ecco turned to me. "Suppose one hole will cause the gourd to drain in n minutes and two holes will cause the gourd to drain in the minute below half of n (for example, if it takes n=20 minutes to drain with one hole, then

Dennis, a professor of computer science at New York University, is the author of The Puzzling Adventures of Dr. Ecco *(Dover, 1998),* Codes, Puzzles, and Conspiracy *(W.H. Freeman & Co., 1992),* Database Tuning: A Principled Approach *(Prentice Hall, 1992), and (coauthored with Cathy Lazere)* Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists *(Springer Verlag, 1998). He can be contacted at DrEcco@ddj.com.*

COME TEST YOUR APTITUDE AS A PROGRAMMER AT SD '99 EAST IN WASHINGTON D.C.—TEST OFFERED ON NOVEMBER 10TH AND 11TH.

One of seven is a good hire...But which one?

Beat the odds!

With Walden Personnel Testing & Training specializing in pre-employment skills-testing for high-tech positions, you can reduce hiring errors!

Walden tests scientifically measure the aptitude, proficiency and knowledge of your job applicants. Once the test is scored by our professionals, you will quickly receive a detailed evaluation of each candidate. The tests are in-depth, job-related, closely simulate on-the-job assignments and predict job success.

Come test your aptitude as a programmer! Sign up to take the Walden Programmer Analyst Aptitude Test FREE OF CHARGE! As part of *DDJ's* Career Development program, you can take this valuable 1-hour aptitude test at no charge. If you've ever wondered how you score as a programmer, here's your chance! To sign up visit the Walden Personnel Testing & Training booth on the show floor at SD '99 East! For info on the test, check out **www.waldentesting.com.**

WALDEN PERSONNEL TESTING & TRAINING INC.

Call TOLL FREE!
800-361-4908
email:
tests@waldentesting.com

Dr. Dobb's JOURNAL SOFTWARE TOOLS FOR THE PROFESSIONAL PROGRAMMER

SD'99 is the largest independent event dedicated to the development professional. Condense months of research and education into a few short days at the SD'99 Conference with our classes and tutorials which give you essential, practical information from the most qualified educators in the industry. Also, test drive the newest tools from over 150 vendors on the expo floor. Your expo pass allows you to participate in panel discussions, attend technical sessions, and join in special events and parties.

SD|99 software DEVELOPMENT

Software Development '99 East

November 8-12, 1999

Washington Convention Center

Washington, D.C.

http://www.sdexpo.com

(continued from page 123)
two holes requires the integer below (20/2)=9 minutes; if it takes n=21 minutes with one hole, then with two holes it takes the integer below (21/2)=10 minutes). Are there any numbers n that make marking minutes impossible?"

Reader: Which values of n *would be troublesome, if any? If none, then map out a procedure that can always mark minutes, assuming you have enough gourds.*

Last Month's Solution

Best val: 1801 and deviation for that ordering is: 64

Object	Alice	Brad	Carla
27	18	33	61
14	49	35	52
7	7	6	49
19	42	13	1
25	64	1	124
6	17	10	17
16	14	7	36
5	2	16	0
2	4	29	33
23	75	9	11
12	3	18	5
29	34	29	3
15	15	22	101
11	41	26	5
13	45	22	11
10	36	2	30
30	37	12	11
3	131	72	62
24	10	33	44
4	9	28	42
17	34	31	47
8	56	1	45
21	30	4	20
22	11	59	39
9	80	4	27
28	7	11	28
18	4	87	24
26	12	19	19
1	71	203	21
20	42	158	32

Deviation of 1 and highest value for this deviation is: 1780

Object	Alice	Brad	Carla
6	17	10	17
18	4	87	24
21	30	4	20
1	71	20	321
30	37	12	11
8	56	1	45
28	7	11	28
13	45	22	11
11	41	26	5
16	14	7	36
15	15	22	101
10	36	2	30
3	131	72	62
25	64	1	124
4	9	28	42
14	49	35	52
29	34	29	3
24	10	33	44
26	12	19	19
2	4	29	33
22	11	59	39
27	18	33	61
5	2	16	0
9	80	4	27
23	75	9	11
12	3	18	5
7	7	6	49
19	42	13	1
20	42	158	32
17	34	31	47

Reader Notes

Dr. Ecco's solution for "Laser Shuttles" (*DDJ*, August 1999) was not the best possible. Fortunately, several clever readers found better ones. In the best solutions, no astronaut will be required to take more than three hops to go from site to site and the total number of edges is 45.

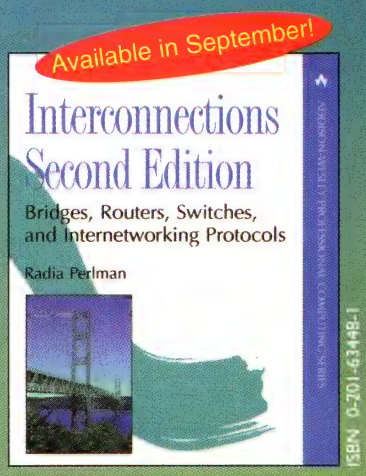
- Alan Dragoo's solution is a nice example (http://members.aol.com/adragoo/laser.jpg).
- Burghart Hoffrichter pointed out: "By counting the number of horizontal and vertical edges in each rectangle and one diagonal per rectangle and the sides of the triangles, one sees that 45 is the maximum number of nonintersecting edges."
- Brian Nelson thought that deregulation might bring about a reduction in the number of edges.
- Kees Rijnierse suggested that putting the stations slightly off the grid might make even better solutions possible, but did not know how.
- Denis Birnie of Transportation & Traffic Systems Ltd (Christchurch, New Zealand) suggested several other optimization criteria. Among them: Is there a 4-hop-to-anywhere network such that every journey can be completed by at least 2 routes with only the start and end stations in common? If not, is there a 4-hop primary and 5-hop secondary network?

Other readers who demonstrated 45 edge/3-hop designs included: C. Brock Rooney, Pearl Pauling, Rodney Meyer, Peter Scholz, Brian Nelson, Tapani Lindgren, Todd Ginsberg, Ted Alper, Mike White, Vernon R. Brown, Stephen A. Slezic, Kiia Kallio, Tom Rokicki, Matthew D. Ahl, Andy W. Desplenter, Mark Taylor, Kees Rijnierse, Forest Wilkinson, and Doug Newhort (who minimized the average length as well, http://members.home.net/dmewhort/3hub.htm).

Still others demonstrated 45 edge/4-hop designs: James Waldby, Christopher W. Curtis, Burghart Hoffrichter, Jason Patterson, Douglas Price, Magne Oestlyngen, John Porter, Bob Morton, Greg Smith, Kees Rijnierse, Denis Birnie, Carey Bingham, Kevin Fox, Steve Bourassa, David Stevenson, Jeffrey A. Hallman, and John Grant.

DDJ

Dr. Radia Perlman's Classic, *Interconnections*, Now Thoroughly Updated!



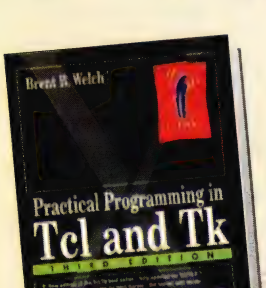
This new edition provides the latest information on important areas of networking: routers, bridges, switches, hosts, and protocols.

"If network architects and standards developers absorb this book perhaps the next generation of networking will be less complex and more effective."
–Tony Lauck

Praise for the First Edition:
"This is one of the best sources of information on the design of network routing protocols that exists. The book is indispensable..."
–Thomas H. Ptacek, Sonicity, Inc.

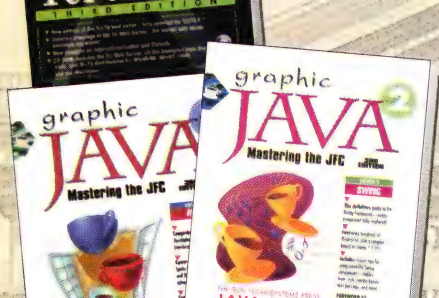
For more information or to order:
1-800-824-7799
Addison-Wesley
www.awl.com/cseng/

The DEFINITIVE CLASSICS for the New Millennium ... FROM PRENTICE HALL PTR



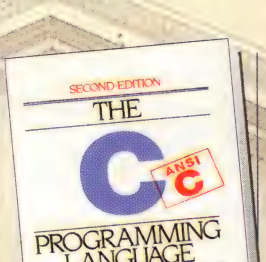
PRACTICAL PROGRAMMING IN TCL AND TK
Third Edition
Welch • 2000, Paper, 0-13-022028-0

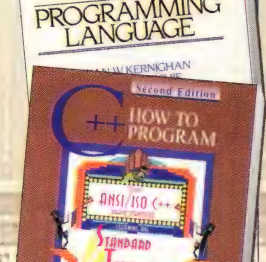
GRAPHIC JAVA 2, Third Edition
Volume 1: AWT
Geary • 1999, Paper, 970pp, 0-13-079666-2
Volume 2: Swing
Geary • 1999, 1600pp, Paper w/ CD-ROM, 0-13-079667-0



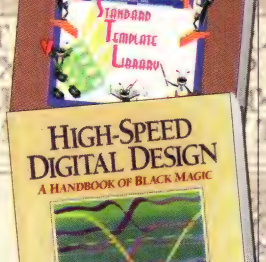
JUST JAVA 2, Fourth Edition
van der Linden • 1999, Paper Bound w/CD-ROM, 608pp, 0-13-010534-1

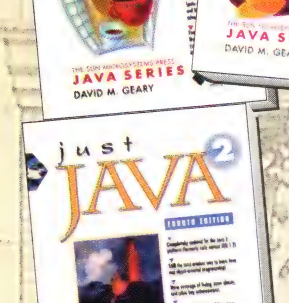
C PROGRAMMING LANGUAGE
Second Edition
Kernighan • 1989, Paper, 274 pp, 0-13-110362-8





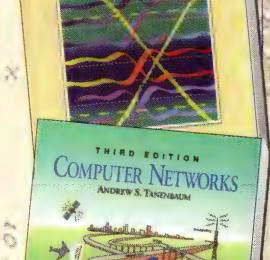
C++ HOW TO PROGRAM,
Second Edition
Deitel / Deitel • 1998, Paper, 1130pp, 0-13-528910-6



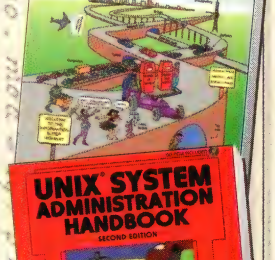


HIGH SPEED DIGITAL DESIGN
A Handbook of Black Magic
Johnson • 1993, Cloth, 384 pp, 0-13-395724-1

COMPUTER NETWORKS, Third Edition
Tanenbaum • 1996, Cloth, 848 pp, 0-13-349945-6

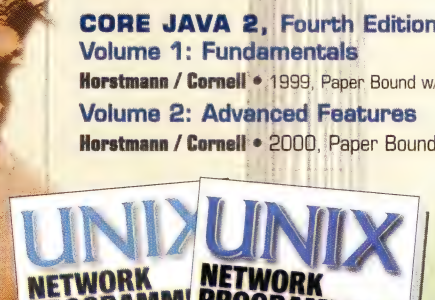


UNIX SYSTEM ADMINISTRATION HANDBOOK
Second Edition
Nemeth / Snyder/ Seebass/ Hein
1995, Paper Bound w/CD-ROM, 780 pp, 0-13-151051-7



UNIX NETWORK PROGRAMMING, Second Ed.
Volume 1: Networking APIs - Sockets and XTI
Stevens • 1998, Cloth, 1240pp, 0-13-490012-X
Volume 2: Interprocess Communications
Stevens • 1999, Cloth, 592 pp, 0-13-081081-9

CORE JAVA 2, Fourth Edition
Volume 1: Fundamentals
Horstmann / Cornell • 1999, Paper Bound w/CD-ROM, 742pp, 0-13-081933-6
Volume 2: Advanced Features
Horstmann / Cornell • 2000, Paper Bound w/CD-ROM, 0-13-081934-4



AVAILABLE AT FINE TECHNICAL BOOKSTORES NEAR YOU, INCLUDING:

California:
Computer Literacy
Digital Guru Bookshop
Fatbrain.com
San Diego Technical Bo
Stanford Bookstore

Massachusetts:
Bookpool
Quantum Books
Softpro Books

New York:
Total Information Inc.

Oregon:
Powell's Technical Boo

Magnet Store
Available at your local technical bookstore.
FOR THE STORE NEAR YOU VISIT OUR WEBSITE!

Direct Orders: 800-288-4745
Corp. Orders: 800-382-3419

www.phptr.com

The Computing Gods have spoke

A Joke Too Far

Gregory V. Wilson

We've all been there, I suppose. You tell a joke, and everyone laughs, so you tell another, and another, until suddenly you realize that you've gone a joke too far. Unfortunately, a lot of software is like that (Emacs, anyone? Or Windows 2000?)—and so are a lot of books.

A case in point is *Mr. Bunny's Big Cup o' Java*, by Carlton Egremont III, the sequel to *Mr. Bunny's Guide to ActiveX*. The latter (reviewed in our April 1999 issue) was hilarious, but *Mr. Bunny's Big Cup o' Java* falls completely flat, as if the author had used up all his good jokes. If you have the money to spend, buy a second copy of *ActiveX* to replace the one your office mates have "borrowed," rather than *Big Cup o' Java*.

Alan Cooper's second book, *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*, is definitely not "one too far"—it makes too many important points about why software is hard to use for that to be the case. Cooper is the father of Visual Basic, and by all accounts (including the ones quoted in the front pages of his book) he is a very good user-interface designer. I learned a lot from *About Face*, and hoped that this one would be equally insightful and iconoclastic.

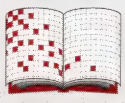
Well, iconoclastic it certainly is. Take the diagram on page 35, for example. It shows "power users," "computer literate users," and "naive users" stacked on top of each other, but is captioned "Euphemism Pyramid." As the accompanying text explains:

> "Naive user" is industry code for "stupid, incompetent user." And certainly these people are made to feel stupid, but they are not. It is the bad design of the interaction that is at fault. The apologists in the computer industry dismiss naive users as not significant, but that flies in the face of good sense. Why would a vendor write off the lion's share of the market? Because it removes the blame for failure from the executives and software engineers and places it squarely onto the shoulders of the innocent users...The phrase "computer literate user" really means the person has been hurt so many times that the scar tissue is thick enough so he no longer feels the pain.

Bravo, Mister Cooper—it's refreshing to hear someone finally say that the emperor's butt is showing. Cooper argues that most software is hard to use because most engineers are disdainful of the way most people think. The only way to fix what's fundamentally broken in so much of today's software is to give interaction designers as much real authority in the product development process as engineers.

Cooper's points are important, and he has a clearer view of our industry and profession than most writers. However, I think that the book's occasional glibness and one-sidedness weaken his case. For example, the two-dollar psychoanalysis of nerds on page 103 might be an accurate description of some engineers, but when applied to the profession as a whole, it becomes the kind of grossly misleading generalization that I associate with paradigm-shift-per-second marketing gurus. More troubling is the fact that he never discusses how the interaction-centered methods he advocates can go wrong. I understand how frustration with current practices can turn into stridency, but I think the major audience for this book would find even-handed exposition more persuasive than naked advocacy. I don't believe anyone's method works all of the time, and when someone suggests (even by omission) that theirs does, it makes me suspicious of everything else they've said. Despite this, I think that *Inmates* is an important book, and I hope that Cooper will turn his hand to writing a textbook one day.

Next in this month's pile are *Ready-to-Run Java 3D*, by Kirk Brown and Daniel Petersen, and *The Java 3D API*



Mr. Bunny's Big Cup o' Java
Carlton Egremont III
Addison Wesley Longman, 1999
115 pp., $14.95
ISBN 0-201-61563-0

The Inmates Are Running the Asylum
Alan Cooper
Sams, 1999
261 pp., $25.00
ISBN 0-672-31649-8

Ready-to-Run Java 3D
Kirk Brown and Daniel Petersen
John Wiley & Sons, 1999
400 pp., $44.95
ISBN 0-471-31702-0

The Java 3D API Specification
Henry Sowizral, Kevin Rushforth, and Michael Deering
Addison Wesley Longman, 1998
482 pp., $39.95
ISBN 0-201-32576-4

Advanced CORBA Programming with C++
Michi Henning and Steve Vinsoki
Addison Wesley Longman, 1999
1083 pp., $54.95
ISBN 0-201-37927-9

Inside the JavaOS Operating System
Tom Saulpaugh and Charles Mirho
Addison Wesley Longman, 1999
184 pp., $29.95
ISBN 0-201-18393-5

Greg is the author of Practical Parallel Programming *(MIT Press, 1995), and coeditor with Paul Lu of* Parallel Programming Using C++ *(MIT Press, 1996). Greg can be reached at gvwilson@interlog.com.*

Specification, by Henry Sowizral, Kevin Rushforth, and Michael Deering. Both books are about Java 3D, the 3D graphics entry in the JavaMedia suite of interfaces. As with the rest of Java, there is little in the library that is actually new. Instead, it repackages established technology (primarily OpenGL) for use in Java. The library includes the usual graphics primitives and transformations, along with behaviors, handlers for input devices, picking, and so on.

The first major difference between the two books is how much context each provides. Almost all of *The Java 3D API Specification* is devoted to a method-by-method description of the library's contents, while *Ready-to-Run* spends a bit more time showing how those methods can be stitched together. I wouldn't recommend trying to find your way around the former book unless you are already familiar with 3D computer graphics, but if you are, it's a handy reference and is refreshingly free of typos.

Which brings us to the other big difference between the two books: their quality. *Ready-to-Run* is, quite frankly, sloppy. The first figure is unreadable because of gratuitous 3D chart junk, matrices are laid out as ASCII art (honestly, how hard would it have been to make them look like proper mathematics?), Arthur Clarke is misquoted, someone's spellchecker has turned the "Locale" class into "Local" sporadically, and some of the listings include Sun's lengthy disclaimers verbatim, along with the source code that the reader is supposed to be interested in. And do we really need yet another slapdash description of how to nest applets in HTML documents for different browsers?

I really wish *Ready-to-Run* had been a better book, as I don't think novices will be able to make sense of the *API Specification*. As hardware graphics accelerators are getting cheaper, and 3D graphics are growing more and more accessible, I hope its authors will try again.

Further down in my pile is Michi Henning and Steve Vinoski's *Advanced CORBA Programming with C++*. I've been waiting for this book for a while, and I can honestly think of nothing to complain about except its inordinate fondness for acronyms. It is well written, comprehensive, and contains lots of useful bits of code for its readers to study and copy. It isn't particularly easy going, but that is merely an honest reflection of the complexity of its subject. The book does not merely describe CORBA, it explains it—why certain features were necessary compromises, for example, or what errors can be made when this or that feature is misused. I think this book is going to be as essential a part of CORBA programmers' lives as Stevens's *UNIX Network Programming* has been for everyone who has had to wrestle with TCP, UDP, and their ilk. And wonder of wonders, its index is actually useful...

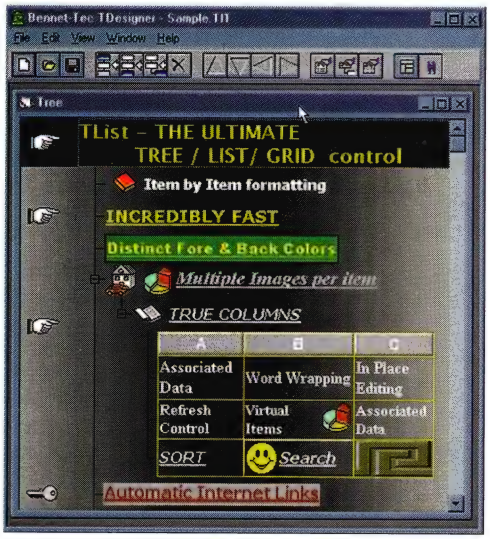
Finally, there is *Inside the JavaOS Operating System*, by Tom Saulpaugh and Charles Mirho. Jointly designed by Sun and IBM to be a universal platform for Java programmers, the JavaOS specifies services, device-driver interfaces, memory management, interrupts, a microkernel, miscellaneous events, and a boot interface. The book surveys all of these, discussing both how they fit together, and how they can be used. Except for a couple of paragraphs of marketing spiel in the introduction, the book is relatively hype free (at least by Java standards). If JavaOS had taken off, this would probably have become the equivalent of Chappell's *Understanding ActiveX and OLE*.

Now, have you heard the one about the three little Ethernet hubs?

DDJ

TList 4 - Tree/List/Grid

Mention DDJ & receive $25 off your order



Bennet-Tec - Top Quality Suppport
TList - First in Speed & Reliability

TList 4 is the fastest, most flexible Tree control available. Add 10,000 nodes per second, load 100,000 node data file in 5 seconds, or use Virtual Items for even higher performance.

- **Distinct formatting by list item, hierarchic level, column, or cell**
- **True Column/Grid support - images & text in each cell**
- **Search or Sort on any column or associated item data**
- **Internet HyperLinks • In-Place Editing • ToolTips**
- **Multiple Selections • WordWrapping • Transparent background**
- **Clipboard Support • Category-based images and hidden items**
- **Fast File I/O - save or load a complete tree or individual branches**
- **Multiple ItemData values for each item**
- **Ease of Use - a single common syntax for Lists, Trees, Grids**
- **Includes TDesigner application - design trees & grids without code**
- **Evaluation Kit on Internet**

Other Bennet-Tec Controls:

- **ALLText HT/Pro-HyperText Enhanced RTF**
- **MetaDraw - Graphic HotSpots, Annotation & Diagramming**

Bennet-Tec
Information Systems, Inc

Bennet-Tec Information Systems
50 Jericho Tpk., Jericho, NY 11753
TELE: (516) 997-5596, FAX: (516) 997-5597
Email: Info@Bennet-Tec.com
http://www.Bennet-Tec.com

The Advertiser Index

Visit Our Advertisers' Web Sites Today!

Advertisers Name	Page #
a2i www.axsnow.com	105
Abraxas Software, Inc www.abxsoft.com	38
Academic Press www.mkp.com	111
Accusoft www.accusoft.com	101
Addison Wesley www.awl.com/cseng/	125
Aladdin Knowledge Systems www.aks.com/dobbs	13
Amazon.com www.amazon.com/computers	115
American Cybernetics www.multiedit.com	71
Artisan Software Tools www.artisansw.com	69
Be, Inc www.be.com	93
Bennet-Tec Information Systems www.bennet-tec.com	128
BlueWater Systems www.bluewatersystems.com	68-82
Boeing Recruitment www.boeing.com/employment	S3
Cambridge University Press www.cup.org	38
Carnegie Mellon University www.distance.cmu.edu/moreinfo/	92
ComputerJobs.com www.computerjobs.com	35
Computer Associates www.cai.com/jasmine/dev	41
Compuware, NuMega Labs www.compuware.com/numega	43
Comtrol Corp www.comtrol.com	70
The Corliolis Group www.coriolis.com	90
DDJ Numerics CD-ROM www.ddj.com/cdrom/	68 & 133
DDJ CD-ROM Library www.ddj.com/cdrom/	130-131
DDJ Programmer's Resource www.ddj.com	134
DDJ CD Release 7 www.ddj.com/cdrom/	S11
DDJ Alternative Languages www.ddj.com/cdrom/	S13&S21
DDJ Algorithms CD-ROM www.ddj.com/cdrom/	S23
DDJ Cryptography CD-ROM www.ddj.com/cdrom/	S18
DDJ Python CD-ROM www.ddj.com/cdrom/	S13
Dell Computer www.dell.com/smallbiz/dev	16-17
Diab-SDS www.diabsds.com	85
Dice.com www.dice.com	122
Dice.com www.dice.com	S14-S15
Esmertec, Inc www.esmertec.com	75
FairCom www.faircom.com	6
Geodesic Systems, Inc www.geodesic.com	104
Gimpel Software www.gimpel.com	52
Global Majic Software www.3DLinX.com	44
Globetrotter Software www.flexlm.com	51
Hi-Tech Software www.htsoft.com	73
IBM www.ibm.com/developerworks	57
IBM www.ibm.com/software/data	118
IBM www.ibm.com/whywork	S1
Installshield Software Corp www.installshield.com/peace	9
Intel Corp developer.intel.com/vtune	23
Intersystems www.licensetospeed.com	106
Intuitive Systems www.optimizeit.com	116
KL Group, Inc www.klgroup.com/widgets	12
Kenonic Controls Limited www.crypkey.com	121
Keystone Learning Systems www.keystonetraining.com	137
Lead Technologies www.leadtools.com	54
LearningPatterns.com www.learningpatterns.com	48
Linux Journal www.linuxjournal.com	92
Linux World Expo www.linuxworldexpo.com	143
MacMillan Publishing www.informit.com	7
MacMillan Publishing www.informit.com	S17
Metaware, Inc www.metaware.com	83
Metrowerks, Inc www.metrowerks.com	87
MFWeb www.mfweb.com	117
MKS www.mks.com/interop	91
Microsoft msdn.microsoft.com/vstudio/win2000	C2-1
Microsoft msdn.microsoft.com/vbasic/vblang	30-31
Microsoft Press www.mspress.microsoft.com	99
Microway www.microway.com	72
ModComp, Inc www.modcomp.com/viewmax	59
Motorola Corp www.employment.motorola.com	97
Northwoods Software www.trulyvisual.com	140
Novell developer.novell.com/nds/	89
Novell www.developer.novell.com/education	S9
Objectivity, Inc www.objectivity.com/lever	61
On Time www.on-time.com	80
Opus Software www.opussoftware.com	114
Parasoft Corp www.parasoft.com/believe.htm	11
Parasoft Corp www.parasoft.com/story.htm	47
Perforce Software, Inc. www.perforce.com	19
Pervasive Software www.pervasive.com	33
The Portland Group www.pgroup.com	102
Prentice Hall www.phptr.com	126
Product Debut	140
Programmer's Marketplace	141-142
Programmer's Paradise www.pparadise.com	14-15
QNX www.qnx.com	67
Quadron www.quadron.com	140
Raima www.raima.com/java	77
Real-Time Innovations, Inc www.rti.com	81
Research Systems www.rsinc.com/go	65
Rogue Wave Software www.roguewave.com/ad/catch	2
Rogue Wave Software www.roguewave.com/ad/studio	3
RSA Data Security www.rsasecurity.com/win	C3
RSA Data Security www.rsasecurity.com	20
RSA Data Security www.rsasecurity.com	29
Sandstone Technology www.sand-stone.com	58
SD '99 East www.sdexpo.com	138-139
Seagate Software www.seagatesoftware.com/vb	37
Sequoia Software www.xmlindex.com/dd	95
Soffront www.soffront.com	45
Softel vdm www.softelvdm.com	39
Software FX, Inc www.softwarefx.com	110
Starbase Corp www.starbase.com	27
State Farm Insurance Co www.statefarm.com/careers/	S7
SubSystems, Inc www.subsystems.com	141
Sun Microsystems www.sun.com	25
Symantec Corp. www.visualcafe.com/vc3/javaposter	C4
Sysnetics www.sysnetics.com	140
Teamshare, Inc www.teamshare.com	112
TechExcel www.devtrack.com	109
Technetcast.com www.technetcast.com	S24
Together Software Corp www.togethersoft.com	63
Tower Concepts www.tower.com	120
Transcender www.transcender.com	39
Transitive Dynamics, Inc www.TranDyne.com	53
Transitive Dynamics, Inc www.TranDyne.com	55
Treck, Inc www.treck.com	92
Unipress Software www.unipress.com	55
Walden Personnel Testing www.waldentesting.com	124
WIBU-SYSTEMS www.wibu.com	140
Wind River Systems www.windriver.com/html/toysdrd.html	79
Wolfram Research www.wolfram.com/warp/ddj	S19
Zero G Software www.ZeroG.com	114

Sales Manager/West
Stan Barnes
650-655-4190
sbarnes@mfi.com

Randy Byers
650-655-4301
rbyers@mfi.com

Hassan Halevy
650-655-4191
hhalevy@mfi.com

Southwest/Texas
Ron Cordek
949-574-0313
rcordek@mfi.com

Gabriel Rogol
650-655-4168
grogol@mfi.com

Northwest
Michael Beasley
650-655-4304
mbeasley@mfi.com

Tom Hudner
650-655-4323
thudner@mfi.com

Associate Publisher/East
Brenner Fuller
603-746-3057
bfuller@mfi.com

Mid-Atlantic/Southeast
Michael Kelleher
561-785-6322
mkelleher@mfi.com

Linda Guyette
603-924-5971
lguyette@mfi.com

NewEngland/NorthCentral
Philip Marshall
978-499-4933
pmarshall@mfi.com

Marla Woods
803-806-8431
mwoods@mfi.com

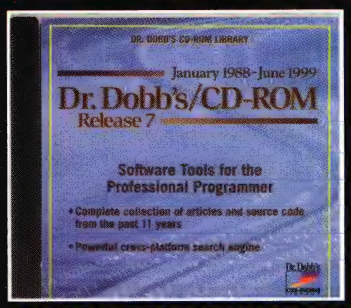
Software Careers
Elizabeth Doherty
650-655-4193
edoherty@mfi.com

Fax: 650-358-9749
www.ddj.com

The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

Visit our web site for all the products from *Dr. Dobb's CD-ROM Library:*

www.ddj.com/cdrom/



Dr. Dobb's/CD Release 7 CD-ROM

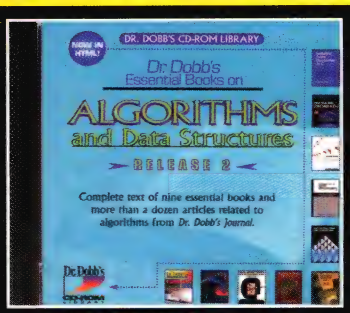
HTML!

Dr. Dobb's/CD Release 7 contains over 11 years of cutting-edge computer information. Including over 132 issues of *Dr. Dobb's Journal* plus all the issues of *Dr. Dobb's Sourcebook* on one CD-ROM! Every Programming Paradigms, every Algorithm Alley, every column, every bit of source code from January '88 to June '99, all at your fingertips.

Dr. Dobb's/CD Release 7 brings together more than eleven years worth of perspectives and expertise from the most respected programmers in the industry. Since *Dr. Dobb's Journal* is written by programmers, for programmers, you receive the news and views from inside the industry. *Dr. Dobb's/CD Release 7* offers both long time and first-time readers an invaluable programming resource.

NEW RELEASE! PRICE! $69.95

$29.95 Upgrade available to prior owners! To upgrade call: 800-822-1162



PRICE! $69.95

Dr. Dobb's Essential Books on Algorithms and Data Structures Release 2 CD-ROM

HTML!

There's nothing more fundamental to programming than algorithms and data structures. Your CD-ROM includes nine of the most important books ever written for programmers about the subject.

These books give you proven and trusted ways to analyze and solve problems in C, C++, and other languages. Take advantage of the research and testing that went into developing the algorithms and data structures included on this CD-ROM!

Master practical programming techniques. Ensure that your programs are understandable, maintainable, and efficient–even as they grow in size. The right data structure can make your operations simple and efficient, the wrong one can make an operation cumbersome and slow.

Powerful cross-platform search engine locates information fast across all books!

INCLUDES THESE BOOKS:

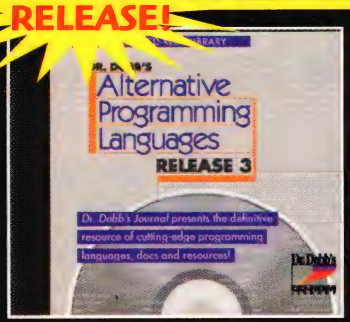
Chosen by the Editors of *DDJ:*

1. *Data Structures: From Arrays to Priority Queues* by Wayne Amsbury
2. *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest
3. *Information Retrieval: Data Structures and Algorithms* edited by William B. Frakes and Ricardo Baeza-Yates.
4. *Fundamentals of Data Structures* by Ellis Horowitz and Sartag Sahni
5. *Reliable Data Structures in C* by Thomas Plum
6. *Data Structures and Algorithm Analysis in C* by Mark Allen Weiss
7. *Practical Data Structures in C++* by Bryan Flamig
8. *Data Structures, Algorithms and Program Style Using C* by James F. Korsh and Leonard J. Garrett
9. *Data Structures and Algorithms* by Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman

Plus more than a dozen articles related to algorithms from *Dr. Dobb's Journal!*

$29.95 Upgrade available to prior owners! To upgrade call: 800-822-1162

NEW RELEASE!



PRICE! $49.95

Dr. Dobb's Alternative Programming Languages Release 3 CD-ROM

HTML!

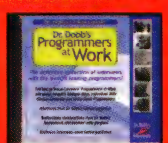
Back by popular demand! *The Alternative Programming Languages Release 3 CD-ROM* is your definitive resource for cutting-edge programming environments, giving you access to the most innovative, solution-providing languages and tools available today.

This new release contains the most comprehensive collection of programming languages ever assembled with over 600 megabytes of data!

The Alternative Programming Languages Release 3 CD-ROM will get you up to speed with languages and docs for Squeak, Dylan, Mozart, Octave, SatherK, Haskell, Lua, EiC, Small Eiffel, Erlang, Modula3, Standard ML of New Jersey and more!

$19.95 Upgrade available to prior owners! To upgrade call: 800-822-1162

ORDER TODAY! 800-992-0549 (U.S. & Canada)



Dr. Dobb's CD-ROM LIBRARY

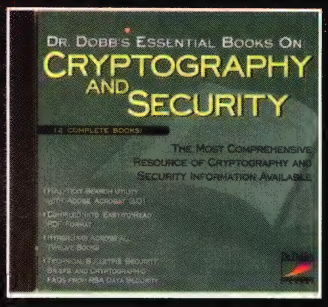
All Other Countries:
785-841-1631

E-mail: orders@mfi.com

Fax Orders: 785-841-2624

Mail Orders:
Dr. Dobb's CD-ROM Library
1601 West 23rd St., Suite 200
Lawrence, KS 66046-2703 USA

Dr. Dobb's Essential Books On Cryptography and Security CD-ROM



The editors of *Dr. Dobb's Journal* have selected the most important books on encryption technology and have compiled it into a resource every programmer must have. This CD-ROM presents both the theory and practice of network security implemented in C, Basic, and other familiar programming languages.

The Essential Books on Cryptography and Security CD-ROM provides detailed documentation and analysis of: Classical and Modern Encryption Algorithms / Private-Key and Public-Key Cryptography / The Data Encryption Standard / Signature Schemes / Block Ciphers and Stream Ciphers / Plus Much, Much, More!

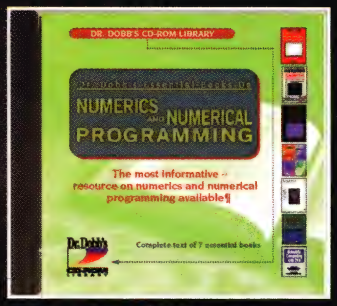
Includes 12 Complete Books! Forward by Bruce Schneier

1. *Applied Cryptology, Cryptographic Protocols, and Computer Security Models* by Richard Demillo
2. *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition* by Bruce Schneier
3. *Contemporary Cryptology: The Science of Information Integrity* edited by Gustavus J. Simmons
4. *Cryptography and Data Security* by Dorothy Denning
5. *Cryptography: A New Dimension in Computer Data Security* by Carl Meyer
6. *Cryptography: Theory and Practice* by Douglas Stinson
7. *Handbook of Applied Cryptography* by Paul C. Van Oorschot, Scott A. Vanstone, and Alfred Menezes
8. *-11 Military Cryptanalysis, Volumes I-IV* by William Friedman
12. *RSA Laboratories FAQ on Cryptography, RSA Laboratories Technical Reports, RSA Laboratories Security Bulletins, and CryptoBytes Newsletter*

PRICE! $99.95

PDF!

Dr. Dobb's Essential Books On Numerics and Numerical Programming CD-ROM



Dr. Dobb's Essential Books on Numerics and Numerical Programming CD-ROM is the answer for anyone who needs solutions for complex numerical equations. Selected for the value of their content by the editors of *DDJ*, these books on CD-ROM are a must have for everyone - - programmer, engineer, statistician, economist, and scientist.

Includes these 8 books:

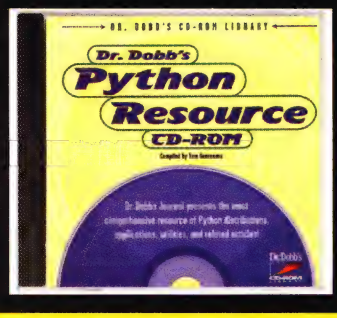
1. *Numerical Recipes in C*, by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery
2. *Scientific Computing with PCs*, by J. C. Nash & Mary M. Nash
3. *Fundamentals of Numerical Computing*, by L.F. Shampine, R.C. Allen Jr., and S. Pruess
4. *Compact Numerical Methods for Computers, Second Edition*, by J.C. Nash
5. *Computing for Scientists and Engineers*, by William J. Thompson
6. *Accuracy and Stability of Numerical Algorithms*, by Nicolas J. Higham
7. *Numerical Methods: Real-Time and Embedded Systems Programming*, by Don Morgan
8. *Elementary Numerical Analysis: An Algorithmic Approach*, by S.D. Conte and Carl de Boor

Compiled into easy-to-read PDF format *Essential Books on Numerics and Numerical Programming CD-ROM* includes these special features: Adobe Acrobat 3.01 / Fast and accurate full-text search utility

PRICE! $99.95

PDF!

Dr. Dobb's Python Resource CD-ROM



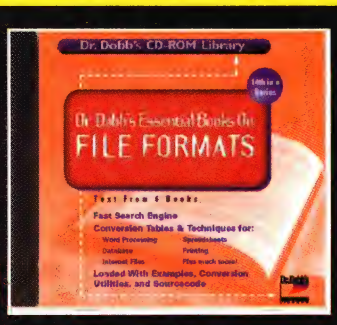
Dr. Dobb's Python Resource CD-ROM is the best available source of Python information you'll find! It includes Python distributions for platforms ranging from Linux to Win32, and utilities and applications for programmers and end users alike. The CD-ROM also includes tutorials, FAQs, and Python articles from leading magazines such as *Dr. Dobb's Journal*, *Web Techniques*, *Linux Journal*, and more!

Python is a powerful interpreted, object-oriented, and portable scripting language that is easily integrated to all of your programming projects.

NEW RELEASE! PRICE! $39.95

HTML!

Dr. Dobb's Essential Books On File Formats CD-ROM



This CD-ROM contains the text from six books, providing the most comprehensive and detailed collection of information on all the important and popular file formats in use today.

Selected by the editors of *Dr. Dobb's Journal*, this CD-ROM contains invaluable information on file formats used for graphics, multimedia, sound, databases, spreadsheets, Windows, the internet, and much more!

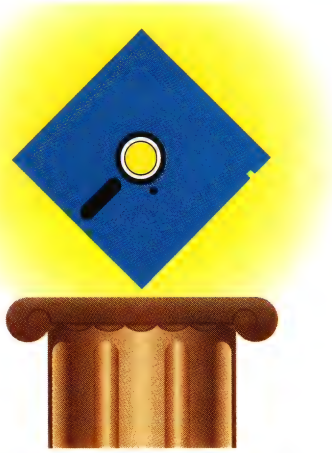
No matter what your programming focus, the *Essential Books on File Formats CD-ROM* discloses the secrets and insider knowledge you need to make your programming efforts instantly compatible with all the major applications — right from the start.

Includes these 6 books:

The File Formats Handbook by Günter Born / *More File Formats for Popular PC Software* by Jeff Walden
Graphics File Formats, 2nd Edition by David C. Kay & John R. Levine / *Internet File Formats* by Tim Kientzle
Inside Windows File Formats by Tom Swan / *File Formats* by Allen G. Taylor

Windows

PRICE! $69.95



Pervasive Software has announced a new version of its Tango 2000 web-development environment. The visual-development studio includes full COM object and JavaBean support, letting you drag-and-drop objects directly into their application files. Tango 2000 also supports XML and saves Tango Application Files in XML format. Tango Class Files let you save Tango code as objects for reuse in other applications. The development environment also includes a Web Analyzer for real-time monitoring of web-site activity. Tango 2000 runs on Windows, Macintosh, and Solaris.

Pervasive Software
12365 Riata Trace Parkway, Building 11
Austin, TX 78727
512-231-6000
http://www.pervasive-sw.com/

Parasoft has released Jtest 3.0, an error prevention and detection tool for Java development. Jtest 3.0 automatically performs static analysis, white-box testing, black-box testing, and regression testing. Jtest checks code against a set of coding standards and offers suggestions on how to improve the quality of the code. It also analyzes the class bytecodes and automatically provides a set of inputs, which can be customized by you. Jtest 3.0 is currently only available for Windows NT, but earlier versions run on Solaris and Windows 98.

Parasoft
2031 S. Myrtle Avenue
Monrovia, CA 91016
888-305-0041
http://www.parasoft.com/

Scenix Semiconductor has announced the SASM SX Cross Assembler, a code development tool that ensures that Virtual Peripheral software modules are completely reusable within any design environment. Virtual Peripheral modules are software implementations of peripheral functions such as A/D converters, UARTs, Caller ID, DTMF generation/detection and other modem functions. The new assembler converts SX symbolic assembly language into object code that is readable by the MS-DOS operating system, which is the standard platform for SX Series development and debugging. The SASM assembler, which supplements an existing assembler from Parallax, Inc. is available from the Scenix web site.

Scenix Semiconductor
3160 De La Cruz Blvd., #200
Santa Clara, CA 95054
408-327-8888
http://www.scenix.com/

Sun Microsystems and IBM have announced the Java Media Framework (JMF) 2.0 API. The JMF 2.0 API features advanced media processing capabilities including audio and video capture, compression and transmission, and support for media types and codecs such as MP3, Flash, and IBM's HotMedia. It also contains an open media architecture to let you access and manipulate various components such as effects, tracks, and renders. The JMF 2.0 API is available through Sun's Java Developer Connection at http://java.sun.com/jdc/.

Sun Microsystems
20450 Stevens Creek Blvd.
Cupertino, CA 95014
http://java.sun.com/

Softel vdm, Inc. has released SftTree/OCX 4.0, an ActiveX tree control for Windows NT/2000 and Windows 95/98 with support for most popular development environments such as Visual Basic, Access, Delphi, and C++ Builder. The new release includes support for virtual lists, a vertical splitter bar between columns, OLE drag-and-drop, digitally signed cabinet files, multicolumn support with cell colors, a background bitmap, and item searching. SftTree/OCX offers hierarchical data displays with advanced features such as multiple columns. SftTree/OCX 4.0 is offered at $299.00 for a single developer license.

Softel vdm, Inc.
318 Tamiami Trail, Suite 14,
Punta Gorda, FL 33950
941-505-8600
http://www.softelvdm.com/

Queue Systems has announced Q-Message, general-purpose messaging software that provides the ability to communicate between applications running on the same platform, applications running on different servers, and applications running on a client-server architecture. Q-Message features a singular interface regardless of the platform, asynchronous programming, an object interface, and does not require socket programming. Q-Message is available for the Open VMS, UNIX, and Windows NT platforms. A server license costs $5000 and $500.00 per client workstation.

Queue Systems
1800 Saint Julian Place
Columbia, SC 29204
803-771-4100
http://www.queuesys.com/

Object Domain Systems has released ObjectDomain 2.5, a UML CASE tool. Features of ObjectDomain 2.5 include support for all nine OMG UML diagrams backed by a true UML meta model; smaller memory footprint; version control and multiuser support via file-based repository; HTML documentation and glossary generation for user models; model searching with regular expression support, and full round-trip for Java. ObjectDomain 2.x will run on any platform that provides a Java Runtime Environment, and supports code generation and reverse engineering for Java, C++, and Python. Existing features also include diagramming with auto-layout, Bezier curves, color support, stereotype icons, and extensive control of presentation aspects. The price of ObjectDomain 2.5 professional addition is $995.00 and the standard edition is $495.00.

Object Domain Systems
1130 Situs Court, Suite 250
Raleigh, NC 27606
919-573-7588
http://www.objectdomain.com/

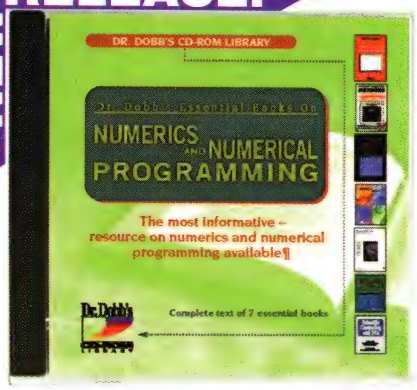
Boxer Software has released Boxer 99, a text editor for Windows NT/98/95. Boxer 99 includes color syntax highlighting, a feature that displays and prints source code for popular programming languages in user-defined colors. Users can modify the reserved words and syntax, allowing Boxer to display any language in any combination of colors and font styles. Boxer 99 lets you open an unlimited number of files, and supports file sizes up to 16 MB and line lengths to 32 KB. Boxer 99 is priced at $59.00 and may be purchased online. Network and multiuser licenses are also available.

Boxer Software
P.O. Box 14545
Scottsdale, AZ 85267-4545
800-982-6937
http://www.boxersoftware.com/

TextPad 4.0 from Helios Software Solutions is a Windows 98/95/NT text editor that supports color syntax highlighting for C/C++, HTML, and Java, as well as free

(continued on page 135)

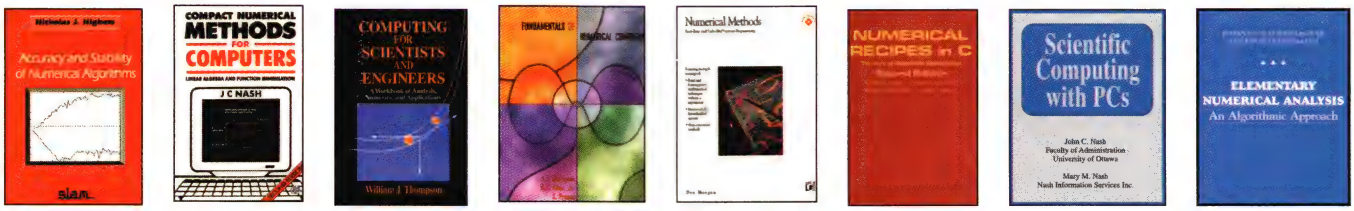
NEW RELEASE!



Dr. Dobb's Essential Books on Numerics and Numerical Programming CD-ROM is the answer for anyone who needs solutions to complex numerical equations. Selected for the value of their content by the editors of *DDJ*, these books on CD-ROM are a must have for all programmers.

$99.95

The Essential Books on Numerics and Numerical Programming CD-ROM gives you the power to search quickly and accurately across the complete text of all eight books.



INCLUDES THESE EIGHT BOOKS:

Accuracy and Stability of Numerical Algorithms, **by Nicolas J. Higham**

Compact Numerical Methods for Computers, Second Edition, **by J.C. Nash**

Computing for Scientists and Engineers, **by William J. Thompson**

Fundamentals of Numerical Computing, **by L.F. Shampine, R.C. Allen Jr., and S. Pruess**

Numerical Methods: Real-Time and Embedded Systems Programming **by Don Morgan**

Numerical Recipes in C, **by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery**

Scientific Computing with PCs, **by John C. Nash**

Elementary Numerical Analysis: An Algorithmic Approach **by S.D. Conte and Carl de Boor**

This comprehensive reference tool will give you:

- Detailed introduction to numerics, numerical analysis, and their applications
- Solutions to numerical problems based on limited memory storage space for data and applications
- Thorough up-to-date treatment of the behavior of numerical algorithms in finite, precision arithmetic, combining algorithmic derivations, and rounding analysis
- Positional number theory, fixed-and floating-point arithmetic
- Multigrid methods for solving partial differential equations
- Spectral analysis on unevenly sampled data
- Calculation of numerical derivatives
- Write efficient, high-speed arithmetic routines, and more!

Compiled into easy-to-read PDF format, *Essential Books on Numerics and Numerical Programming CD-ROM* includes these special features:

- Adobe Acrobat 3.01
- Fast and accurate full-text search utility
- Complete text of all books

Go to our web site for all the products from *Dr. Dobb's CD-ROM Library:*

www.ddj.com/cdrom/

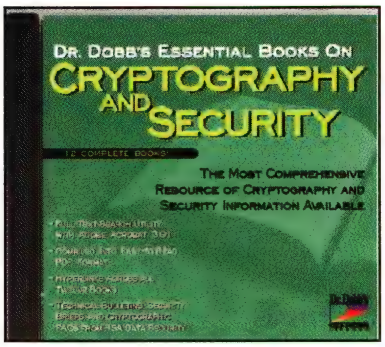
Order Today!
800-992-0549
U.S. & Canada

All Other Countries:
785-841-1631
E-mail: orders@mfi.com
Fax Orders: 785-841-2624

Mail Orders:
Dr. Dobb's CD-ROM Library
1601 West 23rd St., Suite 200
Lawrence, KS 66046-2703 USA

Dr. Dobb's CD-ROM LIBRARY

EXPERT ADVICE WHEN YOU NEED IT MOST!



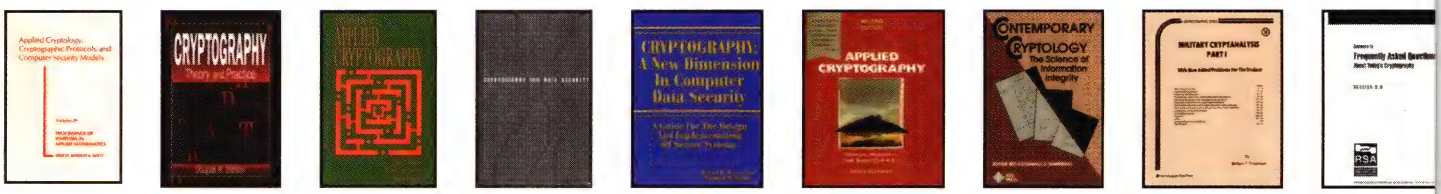
Price: $99.95
Text from 12 books

"Your Product Contains Gaping Security Holes"

-- Bruce Schneier, Contributing Editor

The kind of press coverage you want to avoid, right? Preventing this type of exposure requires extensive planning throughout the entire development process. So where do you turn with questions? A trusted source.

Dr. Dobb's Essential Books on Cryptography and Security CD-ROM



Security and cryptography are among the most important and complex challenges programmers face in today's networked world. That's why the editors of *Dr. Dobb's Journal* have selected and put on CD-ROM twelve essential books all programmers should have on their bookshelf. Written by leading cryptographic and security experts such as William Friedman, Bruce Schneier, Dorothy Denning, Alfred Menezes, and others, *Dr. Dobb's Essential Books on Cryptography and Security CD-ROM* is the definitive resource of cryptographic and security information.

From algorithms for random numbers and hashing, to those for compression and Monte Carlo simulations, *Dr. Dobb's Essential Books on Cryptography and Security CD-ROM* presents both the theory and practice of network security implemented in C, BASIC, and other familiar programming languages.

Add this CD-ROM to your digital bookshelf and possess the single most comprehensive resource of encryption information available.

Includes the following books selected by the editors of *Dr. Dobb's Journal:*

1. *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition,* by Bruce Schneier
2. *Cryptography: Theory and Practice,* by Douglas Stinson
3. *Handbook of Applied Cryptography,* by Paul van Oorschot, Scott Vanston, and Alfred Menezes
4. *Cryptography and Data Security,* by Dorothy Denning
5. *Cryptography: A New Dimension in Computer Data Security,* by Carl Meyer and Stephen Matyas
6. *Applied Cryptography,* by Richard Demillo
7. *Contemporary Cryptology,* edited by Gustavus Simmons

8-11 *Military Cryptanalysis, Volumes I-IV,* by William Friedman

12. *Basic Cryptanalysis,* by U.S. Army

RSA CryptoBytes Newsletter, by RSA Data Security

Now in PDF!

***The Essential Books on Cryptography and Security CD-ROM* provides detailed documentation and analysis of:**

- **Classical and Modern Encryption Algorithms**
- **Private-Key and Public-Key Cryptography**
- **The Data Encryption Standard**
- **Signature Schemes (Digital Signature Standard)**
- **Approaches to Key Distribution and Key Agreement Protocols**
- **Block Ciphers and Stream Ciphers**
- **Generation, Distribution, and Installation of Cryptographic Keys**
- **Communication Architecture Encryption**
- **Authentication Techniques**
- **Applying Cryptography to Pin-Based Electronic Funds Transfer Systems**
- **Measures of Secrecy for Cryptographic Systems**
- **Access, Inference, and Information Controls**
- **Plus Much, Much, More!**

CALL NOW! 800-992-0549

E-mail: orders@mfi.com
Fax: 785-841-2624
Mail:
Dr.Dobb'sCD-ROMLibrary
1601 West 23rd, Ste. 200
Lawrence, KS 66046-2703

Dr.Dobb's CD-ROM LIBRARY

(continued from page 132)
add-ons for Perl, AutoLisp, and Pascal. The size of your documents is limited only by the amount of memory available, and you can edit multiple open documents simultaneously. You can create document classes, associate file names with them, and define different parameters for each document class. Combinations of commands can be saved as keystroke macros and replayed with hot keys. TextPad supports fixed and proportional fonts, as well as double-byte character sets for Asian languages. You can view binary files in hexadecimal format. TextPad 4.0 costs $27.00 and can be purchased online at http://www.textpad.com/.

Helios Software Solutions
P.O. Box 619
Longridge, PR3 2 GW England
44-1772-786373
http://www.textpad.com/

TurboLinux has announced TurboCluster Server, which lets users cluster servers running Sun Solaris and Microsoft Windows NT with Linux. TurboCluster Server runs on open standard Intel and Alpha architecture platforms. It seamlessly integrates into existing networks and virtualizes an IP service by providing a single IP and hostname for a group of servers. These servers each run their own independent copies of the clustered application. TurboCluster Server ships with a suggested list price of $995.00 for two nodes and $1995 for three nodes or more.

TurboLinux
2000 Sierra Point Parkway, Suite 702
Brisbane, CA 94005
650-244-7777
http://www.turbolinux.com/

QNX Software Systems and the Espial Group have announced a PersonalJava solution for set-top boxes, web phones, PDAs, handheld computers, and a variety of other network devices. The tool includes a comprehensive suite of visual design tools, user-interface components, platform products, product APIs, PIM capabilities, and Internet applications. The applications include Espial Escape, a full-featured web browser with a core footprint of less than 175 KB; Ebox, an e-mail client; and Espial Assistant, a complete PIM with an address book, calendar, and appointment manager. The browser adds support for JavaScript, SSL, applets, and printing. The Espial product suite includes DeviceTop, which incorporates an application launcher, support for remote software upgrades and downloads, input device management, virtual keyboards, a multiple application environment, and user/data rights management. Espial also provides the visual design tools Kalos Espresso and Kalos Architect. OEM pricing is available from both Espial Group and QNX Software.

QNX Software Systems Ltd.
175 Terence Matthews Crescent
Kanata, Ontario, K2M 1W8, Canada
800-676-0566
http://www.qnx.com/

Espial Group
200 Elgin Street, Suite 205
Ottawa, Ontario,
K2P 0P7 Canada
800-437-7425
http://www.espial.com/

Code Fusion from Cygnus Solutions combines Cygnus-certified open-source GNU tools with a graphical IDE framework. The Code Fusion IDE features a C, C++, and Java tools product manager, editor, graphical browsers, and the Cygnus Insight debugger. Cygnus Code Fusion supports all major Linux distributions. Code Fusion IDE is priced at $299.00 and is immediately available for online purchase at http://www.cygnus.com/codefusion/.

Cygnus Solutions
1325 Chesapeake Terrace
Sunnyvale, CA 94089
800-294-6871
http://www.cygnus.com/

Lineo (formerly Caldera Thin Clients) has begun shipping an embedded microbrowser product, called Embrowser, which lets you create a customizable Internet browsing interface for embedded devices such as set-top boxes. Targeted at embedded OEMs, Embrowser is intended to be paired with customized hardware, HTML interfaces and backend content designed by the OEM customer. Embrowser currently runs on DR DOS, and a version will also be available on Embedix, Lineo's embedded version of Linux.

Lineo, Inc.
383 South 520 West
Lindon, UT 84042
801-426-5001
http://www.lineo.com/

WoodenChair Software has announced a new suite of namespace management and Javadoc authoring and editing tools. The Utility+ Lite edition supports major integrated development environments (IDEs) such as VisualCafe, VisualAge, and JBuilder. The namespace management tool, Repackager+ Lite, handles package structures, import statements and class names. It updates the IDE project file (*.jpr or *.vep) and all files in the project to reflect the new class name chosen. The Doc+ authoring/editing tool for javadoc documentation allows you to create, remove and edit javadoc tags. Features include auto-format of Javadoc, comment conversion of existing code, autofill of certain tags from source, creation of custom javadoc tags and application of templates created with Doc+. PackageMaps created with RePackager+ can update dependent source in your projects, and Swing PackageMaps is included to update your source to the new javax.swing.* namespace. A standalone version for working with the JDK and text-based editors is also available. Free evaluation copies are available from the company's web site.

Woodenchair Software Corp.
8 Kemsing Court
Markham, Ontario
L3R 4P7 Canada
905-479-2243
http://www.woodenchair.com/

BrainTech has released ODYSEE Development Studio 3.0, a drag-and-drop environment for application development. ODYSEE features an integrated environment with GUI builder; online debugging and optimization; ODYSEE-wrap for integrating C/C++ functions; and a component-based architecture. Such an open architecture has no limits in the type and number of components that can be brought into its environment. A free trial version of ODYSEE Studio is available online at http://www.bnti.com/.

BrainTech Inc.
Unit #102 - 930 West 1st Stre
North Vancouver, BC
V7P 3N4 Canada
604-988-6440
http://www.bnti.com/

Quality Systems and Software (QSS) has released DOORSConnect TestDirector, which links QSS DOORS with Mercury Interactive's TestDirector, allowing software developers to capture end-user requirements and trace them from initial design through final testing. The new tool links the requirements and testing processes. DOORS and TestDirector are synchronized, with a color-coded DOORS view showing the current status of TestDirector tests and defects. DOORS filters can be used to generate various reports that show, for example, only tests that have passed or failed. The package runs on the Windows platform and is available for a single-user license price of $500.

Quality Systems & Software
200 Valley Rd., #306
Mt. Arlington, NJ 07856
973-770-6400
http://www.qssinc.com/

Esker has announced availability of the Alcom LanFax SDK, which lets you embed

fax automation directly into applications. The SDK lets you integrate Alcom LanFax NT with other applications to add automated fax routines into databases, transaction processing systems, document management, imaging, and workflow applications. The SDK is accessible using Visual Basic, C/C++, and Visual C++, as well as scripting languages such as VB Script. This library exposes the programmable fax messaging objects including Alcom LanFax NT phonebook information, fax messages, destination information, attachments, and other fax messaging components. The Alcom LanFax SDK is free and is available via download from http://www.alcom.com/.

Esker Inc.
100 E. 7th Avenue
Stillwater, OK 74074
405-624-8000
http://www.esker.com/

Motorola and emWare have jointly released the NET.08 SDK. Based on Motorola DigitalDNA's 8-bit 68HC908GP32 Flash microcontroller, the NET.08 kit contains a development environment for embedding network connectivity into microcontroller-based devices (like power meters) and building controls and home automation devices without requiring an RTOS or TCP/IP stack in the device. Flash memory on the chip provides flexibility by allowing devices to be debugged, enhanced, or upgraded remotely through the network. The NET.08 software development kit is available for $99.00 from emStore (http://www.emstore.com/motorola/) and contains: Version 3.0 of emWare's EMIT (Embedded Micro Internetworking Technology) device-networking SDK; Netscape and Microsoft browsers; Explorer 4.0; Standard emMicro port to Motorola's 68HCO8 in C source code; emWare hex file used in programming the 68HC08-based device; online general EMIT user's and developer's manuals and user-interface manual; demo applications; online NET.08 specific developer's guide and reference board manuals; Motorola microcontroller reference manuals (on CD-ROM); the reference board including Motorola 68HC908GP32 Flash microcontroller; 9V battery with cable; and an RS232-to-RS232 device cable.

emWare Inc.
6322 South 3000 East, Suite 250
Salt Lake City, UT 84121
801-453-9300
http://www.emware.com/

VMware has announced the release of VMware for Windows NT and Windows 2000. VMware enables PCs to run one or more protected sessions concurrently using one or more operating systems on a single machine. VMware supports a variety of native and Virtual Machine operating systems, including Linux, FreeBSD, NetBSD, and other BSD versions of UNIX and Solaris for Intel processors, as well as Windows 95/98. VMware does not modify the host operating system and does not require rebooting, repartitioning, or reconfiguring the system. It provides a graphical interface for flipping between applications or operating systems running in the background. Users can encapsulate their PC environments and run them on any machine with a simple save and restart procedure. A Pentium processor running at 266 Mhz or faster and a minimum of 96 MB of RAM is recommended.

VMware Inc.
3145 Porter Drive, Building F
Palo Alto, CA 94304
650-475-5000
http://www.vmware.com/

KLGroup has released Jprobe 2.5 Server-side Edition. The Jprobe suite of development tools offers Solaris support, performance tuning capabilities, memory debugging,and thread analysis of Enterprise JavaBeans (EJBs) and servlets running under popular application and web servers. Jprobe 2.5 also supports IBM Visual Age for Java. The tool includes a new web and application integration feature called Server Launch Pad, which automatically configures Jprobe to work with various application servers, either locally or remotely. The Launch Pad currently includes support for IBM WebSphere, NEA WebLogic Server, Java Web Server, Allaire/Live Software Jrun, Sun Servlet Runner, and user-configured Java servers. Pricing for Jprobe 2.5 starts at $499.00.

KLGroup
260 King Street East
Toronto, Ontario
M5A 4L5 Canada
416-594-1026
http://www.klgroup.com/

AuthenTec has announced availability of an Evaluation Kit and a SDK that make it possible for you to develop silicon-based, subsurface fingerprint identification software. The kits include evaluation copies of the company's FingerLoc Personal Identification System, based on its E-Field Technology fingerprint identification technology that reads beneath the surface layer of the skin to the live layer, where the true fingerprint begins. E-Field Technology is not dependent on a finger's surface features, which are constantly affected by changing external factors such as dirt, calluses, and dryness. The Evaluation Kit includes the FingerLoc AF-S1 System along with its biometric evaluation program, which provides enrollment and matching software in both verification and identification modes. Presently, the Evaluation Kit is offered as a standalone PC peripheral with an RS-232 serial interface. The company also plans to release a USB version. The SDK upgrade includes the evaluation kit; development software with APIs; user's guide and programmer's reference manual; and a Windows-based demo for login, file encryption, and password bank. The Evaluation Kit is priced at $995.00 and the SDK is priced at $4595.00. Additional sensors may be obtained for $125 in quantities of 1–10 and for $100 in quantities of 11–100.

AuthenTec, Inc.
P.O. Box 2719
Melbourne, FL 32902-2719
407-308-1300
http://www.authentec.com/

Packet Technologies has announced Sheriff On Board, a firewall on a single chip for application in 10/100 Base-T, PC-LAN NIC, and USB-based networks. The built-in dedicated CPU enables Sheriff On Board to provide application screening, protection against spoofing, real-time scanning of intruders, static filtering, stateful inspection, packet data inspection, network address translation (NAT), and accurate logging and reporting. All incoming packets are parallel processed and analyzed for legitimacy. Processing for both incoming and outgoing packets operates at network speeds of up to 100 Mbits/sec. Sheriff On Board incorporates a web-browser management interface. The system is available in three core versions: a 10/100 base-T to 10/100 base-T firewall on a single chip targeting the cable modem and xDSL modem market, a PCI-LAN NIC firewall on a single chip targeting the mass production market (replacing the standard NIC on the corporate PC and providing LAN connection and a high-level policy-based security solution), and a 10/100 base-T to USB firewall on a single chip targeting new products incorporating the USB interface. Sheriff On Board is based on the company's VLSI-based Reduced Instruction Security Processor.

Packet Technologies Ltd.
6 Hamachtesh Street
Holon 58810 Israel
972-3-558 7001
http://www.packet-technologies.com/

DDJ

www.keystonetraining.com

KEYSTONE LEARNING SYSTEMS

Microsoft Certified Solution Provider

Master C++, Visual C++ & COM in Minutes

Get Real-World C++ Training From Hany Greiss

Dr. Hany Greiss has over 6 years experience as a Senior Technical Instructor in Oracle, UNIX, C++, Windows, Java, Tel, & SQL for PEAC inc, in Ottawa, Ontario. In this course he brings to you his expertise in the specific area of PL/SQL for Oracle 7. Dr. Greiss has 16 years of extensive knowledge as a software developer, senior analyst and consultant. He knows his stuff!

COM/ActiveX

6 Videos, 6 Workbooks
Over 12 Hours of Training

PRICE $499.95 COMPLETE

Course	Price
COM Level 1- Fundamentals of OOP	$89.95
COM Level 2- Fundamentals of COM	$89.95
COM Level 3- COM Features	$89.95
COM Level 4- COM Featrs. DLLs & Registry	$89.95
COM Level 5	$89.95
COM Level 6	$89.95

COM/ActiveX Topics You'll Learn:

Core Concepts of Object Oriented Programming • Core COM Concepts • COM Hands-On Unknown Interface • Unknown Details • Reference Counting • Stand-Alone Components Reference Count Details • Multi-Component DLL's • The Registry • DLL Self Registration Client Using DLL • Component Factories • Implementing IClass Factory • Writing DLL's Support IClass Factory • Automation (IDispatch Interface) • Tools • Smart Pointers Off-the-Shelf Smart Pointers • Wrapper Classes • Server-Side Tools • Basic Principles Automation Servers • Automation Controllers • Active Template • Using COM Objects Automating PowerPoint • MFC Component Creation Building a Component in MFC

We accept C.O.D., Company P.O. or check +

LS Corp.®, WordPerfect®, MS and Microsoft®, Lotus®, Borland®, Aldus® and their products are trademarks and/or registered trademarks f Novell,Inc., Microsoft Corp., Lotus Development Corp., Borland International Inc., and Aldus Corp., respectively.

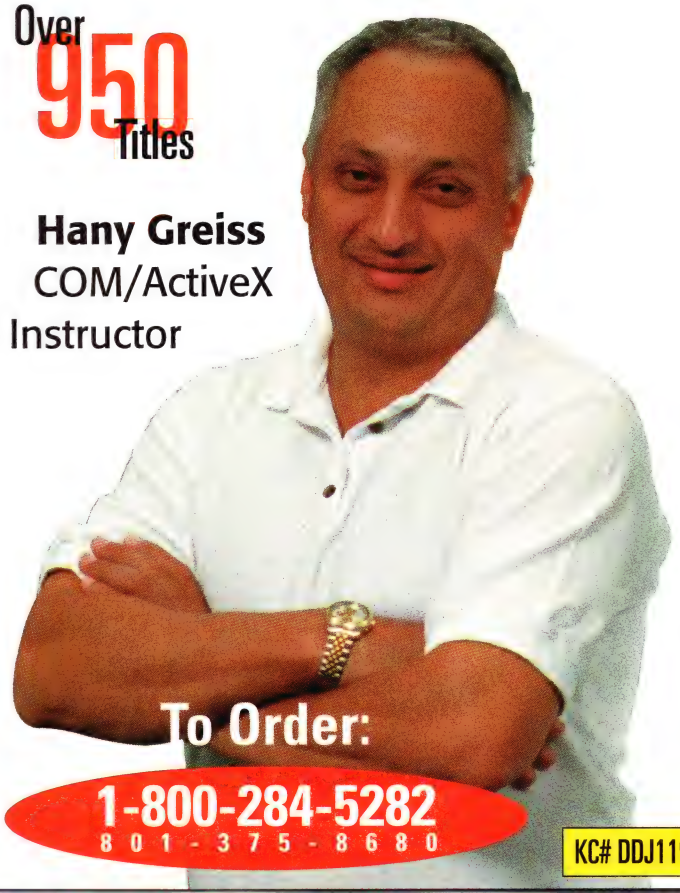
Visual Basic 6.0

17 Videos or CDs, 17 Workbooks
Over 34 Hours of Training

PRICE $1199.95 BEST SELLER

Course	Price
VB 6.0 Getting Started	$599.95
VB 6.0 Power Developer	$699.95
VB 6.0 Component Development	$89.95

Over 950 Titles



Hany Greiss
COM/ActiveX
Instructor

To Order:

1-800-284-5282
801-375-8680

KC# DDJ119

SD'99 Conference and Expo Preview

Conference Tracks

- Building Linux Servers
- C++ Programming
- Component-Based Development for Distributed Systems
- COM&/Windows 2000
- Enterprise Application Integration
- Java Programming
- Methods & Modeling
- User Interface Design
- XML Development

Also featuring our 2-day Management Conference

Visionary Keynotes sponsored by INFO WORLD

OPENING KEYNOTE:
Wednesday, November 10, 10-11am
Bill Joy, Chief Scientist, Sun Microsystems, Inc.
JAVA & JINI: TOWARDS A SCIENCE OF PROGRAMMING

Thursday, November 11, 10-11am
Simon Phipps, Chief XML & Java Evangelist, IBM Corporation
ESCAPING ENTROPY DEATH: THE IMPERATIVE OF XML & JAVA

Special Events & Parties

Focus on hot-button issues and bring tough questions to our expert developers during our special events. Then let off steam at our FUN evening parties.

- Roundtable Discussions
- Design Clinics in Java and C++
- Power Panels
- Birds-of-a-Feather Parties for COM, XML and Java users
- SD'99 Gig & Bites Party
- "You Have the Floor" Party
- Developalooza '99 Party
- Career Development Sessions
- Technical Sessions by

COMPUTER ASSOCIATES® Software superior by design.

Rational unifying software teams

THE EXPO

The year's most anticipated showcase features the full spectrum of development tools, where vendors send their engineers to interact with you. Product launches, free technical seminars and a non-stop series of special events are all part of the action.

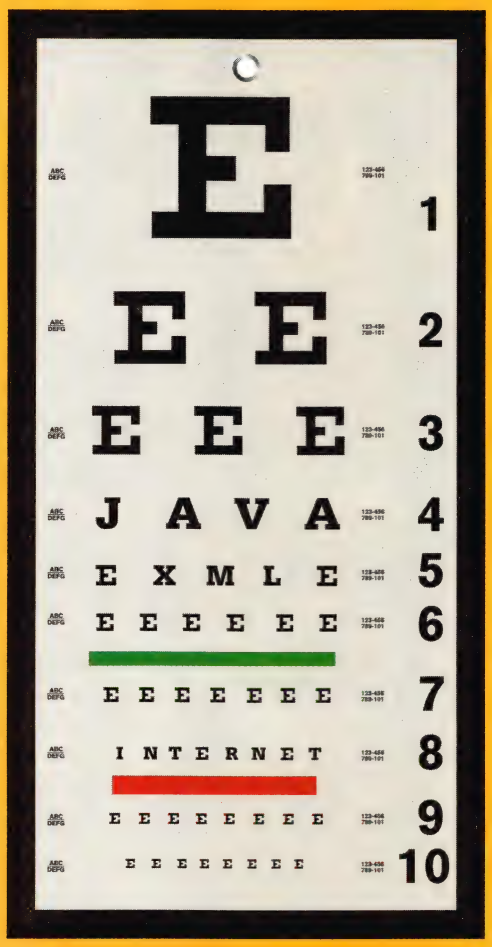
- Targeted Pavilions: IBM Partner Pavilion; XML Pavilion; Distributed Object Computing Pavilion
- Online Lounge: Play networked games or simply network with your peers!
- Exhibitors Include:

AD Experts	LearningPatterns.com	Rogue Wave
Alis	McCabe	Seagull
Computer Associates	Object Design	Sourcecoffsite
Compuware	Parasoft	StarBase
ESRI	Prentice Hall	Teamshare
IBM	Princeton Softech	Technology Exchange
Insight	Rainbow Technology	Template Software
Intraware	Rational Software	Wiz Art Software

Miller Freeman

Sometimes you need the Big Picture to know where to zoom in. That's why Software Development '99 offers both breadth of technologies and depth of content for building your Internet strategy. SD'99 is the only independent conference that gives you all the options—and lets you decide where to focus your vision. That includes not just Internet technologies like XML and Java, but core languages, enterprise development, and platforms such as Linux and Windows 2000. Best of all, SD'99 lets you mingle with your peers at parties and events, or peer into the future with industry visionaries. Because when it comes to software development, opportunities in the future are closer than they appear.

Register for the **VIP Pass** and receive a LIMITED EDITION **SD'99 Fleece Jacket!**



Sharpen Your Internet Focus at SD|99

www.sdexpo.com

CONTACT US FOR
REGISTRATION INFORMATION

415 905 2702 • sd99@mfi.com

SD|99
software
DEVELOPMENT

CONFERENCE NOVEMBER 8—12

EXPO NOVEMBER 9—11

WASHINGTON CONVENTION CENTER
WASHINGTON D.C.

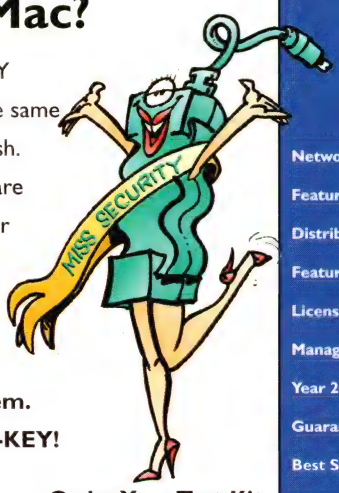
DR. DOBB'S JOURNAL

PRODUCT DEBUT

WIBU-KEY Software Protection

One Solution for PC and Mac?

Yes, the WIBU-KEY USB hardware is the same for PC and Macintosh. Program the hardware at the PC and deliver it with a Macintosh version of software. No problem!



All with one system.
All with WIBU-KEY!

10 YEARS

Network Features, Distribution Features, License Manager, Year 2000 Guarantee, Best Service with every WIBU-BOX.

SYSTEMS 99
19 - 22 OCTOBER, MUNICH
Hall A2 – Booth 533

Order Your Test Kit
1-800-986-6578

WIBU SYSTEMS
Quality the World Trusts

International:
WIBU-SYSTEMS AG
Rueppurrer Strasse 52-54
D-76137 Karlsruhe
tel.: +49-721-93172-0
fax: +49-721-93172-22
email: info@wibu.com

GRIFFIN TECHNOLOGIES
USA, Canada: Griffin Technologies, LLC
tel.: (785) 832-2070 · fax.: (785) 832-8787
email: sales@griftech.com · www.griftech.com

Argentina, Belgium, Canada, Croatia, Denmark, France, Hungary, Japan, Jordan, Lebanon, Netherlands, Spain, Portugal, Syria, USA.

www.wibu.com

Consulting • Mentoring • Training • Placements

SYSNETICS™
Object Oriented Technologists

SYSNETICS™ specializes in Object Oriented (OO) Technology for software architecture, design and development of Distributed Systems.

Based on our exceptionally high standards, people and methods, SYSNETICS™ helps managers minimize risk in the areas of people resources, technical design, architecture and implementation for large-scale OO projects.

- OO Technology: Java,™ C++, Visual C++, Smalltalk
- Distributed Technology: CORBA, COM/DCOM
- OO Methods and Tools: UML, Booch, OMT, Rational Rose, Paradigm Plus

SYSNETICS™
150 153rd Avenue, Suite 300B
Madeira Beach, FL 33708
v: 800-270-6099
e-mail: info@sysnetics.com
www.sysnetics.com

Need high speed?... check out our fast comm cards!

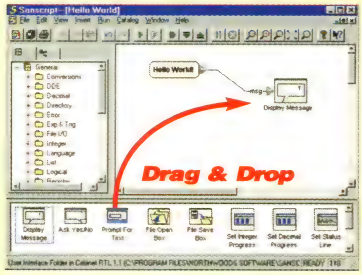
T1/E1, Fractional T1, T3/E3, ISDN, & HSSI speeds.

You'll be able to run Frame Relay, ATM, X.25, PPP, LAP-B, HDLC/SDLC, SNA & custom protocols.

fax 805-966-7630
telephone 805-966-6424
email info@quadron.com

Quadron®
www.quadron.com

Breakthrough in Programming!



- Program with pictures, not code!
- Use with ACT!, Word, Excel, Powerpoint, or any COM-enabled application!

Sanscript ™ - by Northwoods Software

Free Trials

www.TrulyVisual.com

7 GREAT CONTROLS!

- TE Edit Control (Advanced RTF control)
- HTML Viewer/Editor Add-on for TE
- ReportEase Plus (report writer engine)
- SpellTime DLL and dictionary
- FormPlus (form designer/filler)
- Rich Text Grid control and ChartPro

ALL PRODUCTS INCLUDE COMPLETE SOURCE CODE, ARE ROYALTY FREE AND AVAILABLE FOR 16 OR 32 BITS.

Demos: www.subsystems.com

SUBSystems

SUB SYSTEMS, INC.
11 Tiger Row, Georgetown, MA 01833
978-352-9020 Fax: 978-352-9019

Where do you find 116,000 professional software developers every month?

In the DDJ

PRODUCT DEBUT & MARKETPLACE

PRODUCT DEBUT

DR. DOBB'S JOURNAL

MARKETPLACE

C++ LIBRARIES

mime ++

"...a most complete and essential class library"

Licensed by Fortune 500 companies ... ... and around the world!

- √ document object model for MIME
- √ C++ library
- √ fully standards compliant (RFC 822, 2045, 2046, & more)
- √ SMTP, POP, NNTP
- √ source code available

Hunny Software ♦ (301) 948-6999
www.hunnysoft.com/MIMEPP

Fix the Loopholes in C++ Arithmetic

Conventional C++ arithmetic can result in unacceptable rounding errors and even destructive, high-order truncation. Now, you can easily replace numeric datatype declarations (double, int, unsigned long, etc.) with our new patent-pending Base/1 Number Class and improve the accuracy of calculations in any C++ application.

Base/1 Number Class gives you:

- Uncompromisingly exact decimal arithmetic
- Efficient arithmetic and compact representation
- Up to 100 significant digits of precision with decimal point at any desired digit
- Fully documented ANSI C++ source code and examples

Guarantee numeric precision in all your applications.

Visit www.boic.com or call 212-691-7155 to order your copy now for **only $199.**

BOI Base One International Corporation
E-mail info@boic.com • Visit www.boic.com

C++ OR ABRIDGED

Dinkumware® Ltd.
Genuine Software

Dinkum®
C++ or Abridged for Visual C++ V5/6
Libraries & Testing Tools
www.dinkumware.com

COMPRESSION

CRUSHER!™ Data Compression Toolkits

The best cross-platform compression libraries for Win32, Win16, DOS, OS/2, Unix, Macintosh, and embedded systems.

Robust 45-function API compresses buffers, files, archives, disk spanning, encryption, self-extr. EXE's and more.

FREE DEMO Call 1-800-775-1073

DC Micro Development
Tel (678) 442-1623
Fax (678) 442-1819
www.dcmicro.com

CUSTOM SOFTWARE

The Right Choice

- Scientific, Engineering and Industrial software development worldwide.
- 10 year history of successful corporate partnering.
- A Philadelphia 100 company.

SST Scientific Software Tools, Inc.
610-891-1640 • info@sstnet.com

DEVELOPMENT TOOLS

FREE Yourself from the stress, strain and headaches that can accompany custom programming worries!
NAG Numerical Libraries & Visualization tools

NT Windows	C /C++
UNIX	SMP
Solaris	Parallel
Linux	Fortran

• Increase productivity • Reduce costs
• Accelerate development • Cutting edge algorithms • Quality documentation

Email NAG today at info-ddj@nag.com and we will send you information for obtaining your **Free NAG Worrybeads and information on quality, dependable NAG products** to relieve your numeric headaches.

NUMERICAL ALGORITHMS GROUP
630-971-2337 NAG®

Code Co-op™ NEW! Version 2.1

The versatile Version Control System for collaborative development

- Synchronization using email, local network, floppy disk
- Intuitive GUI -- check-in, check-out, synch, visual diff.
- Fully functional trial version available for download

SERVER-LESS VERSION CONTROL
www.relisoft.com

Reliable Software™
Smart Tools for Smart Programmers™

MARKETPLACE

DOCUMENTATION TOOLS

Generate Documentation from your source code with *DocJet*!

Produce HTML, MSHelp, and MSWord documentation from comments in your code - and you won't need to change your commenting style!

You can fine-tune your output with DocJet's WYSIWYG output editor.

FREE TRIAL VERSION!
http://www.tall-tree.com
info@tall-tree.com 512-453-4909

C++ to HTML

- ▶ *New - Version 2.1.*
- ▶ Generates documentation directly from the source code.
- ▶ Extracts comments.
- ▶ User customized reports formats.
- ▶ HTML, WinHelp, RTF.
- ▶ *FREE* working evaluation at www.bbeesoft.com

1-888-646-1933
Bumble Bee Software
P.O. Box 2007
Westford, MA 01886
info@bbeesoft.com

Help & Manual

Full support for Windows98 style HTML HELP

Writing first rate software documentation doesn't have to be hard. Help & Manual is a stand-alone WYSIWYG authoring tool, independent from your programming language and easy to use even for non-programmers. Write your help file once and create various outputs from a single source:

* Windows 98 HTML HELP
* Plain HTML
* Help files for Windows 9x
* Help files for Windows 3.x
* Printed user manuals
* Electronic manuals (RTF)

Get a fully functional evaluation copy at
http://www.ec-software.com

ec Software

EDUCATION

Netscape: Student Section

ADDRESS FOR SUCCESS

http://www.aics.edu

- Earn B.S. and M.S. in Computer Science
- NEW *B.S. program in Information Systems*
- Distance Education
- Object oriented B.S. program
- Approved by more than 275 companies
- Follows ACM/IEEE guidelines
- Thousands of students throughout U.S.

Free catalogue ***1-800-767-AICS***
or ***www.aics.edu***

AMERICAN INSTITUTE COMPUTER SCIENCES
STATE LICENSED AND APPROVED
ACCREDITED *World Association of Universities and Colleges*

GRAPHICS LIBRARIES

New Version

VICTOR

Image Processing Library

Fast **BMP, TIFF, PCX, GIF, TGA, PNG, JPEG**. Adjust brightness, contrast, sharpen, create filters, resize, rotate, +more of single image, multiple images, or any image area; color reduction to optimum, specific, or std. palette; print; scan; crop, combine, compare, blend images.

DOS $199, 16-bit DLL $299, 32-bit DLL $499

Catenary Systems
314-962-7833/fax: 314-962-8037
www.catenary.com/victor
ask for free demo src avail visa/mc/c.o.d.

IMAGING TOOLS

Spectrum Imaging System
Add our full-featured Imaging Software to YOUR Application in Less Than One Day!

- ★ Scans, stores & indexes Images for instant access
- ★ Creates Objects Folders keyed to your Account #'s
- ★ Supports a full range of file types including TIFF, JPEG, GIF, PDF, DOC, XLS, AVI, WAV plus many more
- ★ Includes a Web Object Server to any Browser
- ★ Quickly integrated using ActiveX, DDE, Program CALL, Designer Wizard, or Unix & Mainframe "Screen Scraping"
- ★ Scalable, multi-user, network-enabled, Win95/98/NT

CRT Systems, Inc.
(800) 826-7740 • (803) 327-9030
www.spectrum.crtsys.com/developers

One Stop Imaging Shop for Tools and Apps!

ScanFix® Improve OCR accuracy by 50% or more. 1,000,000+ users worldwide
FormFix® Forms ID and extraction. Allows OCR/ICR to work well on difficult forms
SpectrumFix® Fast color extraction for OCR and more. AIIM show winner
ViewDirector™ The original tool for viewing and decomp, *fast* and **robust**

Demos and information at www.tmssequoia.com
800-867-5250

TMSSequoia
The Imaging Solutions Company

JAVA TOOLS

A Tool Built For Windows Users

The Simple Yet Powerful IDE for Java Development

KAWA

- ➢ JDK Version Independent
- ➢ Plug-In API & Great Debugger
- ➢ Only $59
- ➢ Free 30-day Evaluations

www.tek-tools.com

Bayside Computing, Inc.
Developer Resources

Java Java Java

Beans, Applets, Embedded Info

www.baysidecomputing.com

PRODUCTS WANTED

WANTED
SOFTWARE PROGRAMS
LOOking for partners in profit!

You provide the completed Small Business, SOHO, Financial, Personal Productivity or "how-to" software program. E-Z Legal Software will provide packaging, duplication, marketing and sales personnel to generate royalty checks for you.

With distribution in more than 6500 retail outlets, we have the experience to launch, market and sell your program at retail *at no cost to you!* **Dennis Liptrot: 1-800-822-4566**

384 S. Military Trail, Deerfield Beach, FL 33442
Phone (954) 480-8933 • Fax (954) 480-8906
http://www.e-zlegal.com — dliptrot@e-zlegal.com

E-Z LEGAL SOFTWARE

PROLOG TOOLS

Create Prolog Components

*Diagnose, advise, configure and plan with the Amzi!® LogicServer™ tools & libraries (DLLs) for C/C++, Java, VB, Delphi, Web Servers & more. Win NT 95 3.x & Solaris. Use ODBC, Sockets, Unicode & **new** OOP extensions.*

FREE Evaluation Version!

Amzi! inc. *Email* info@amzi.com
Call +1-513-425-8050 *Fax* 425-8025

www.amzi.com

SECURITY

The Best Try-Before-Buy Copy Protection & Software Licensing

CrypKey is solid, software-only copy protection with a broad range of features to control all aspects of your product's operation. You can offer free one-time trials, turn demos into full multi-user network versions, or sell specific options by phone, fax or email. Sell usage by time, runs, features - the control is all there - easy to implement, and even easier to support; and it's not fooled by reinstallation or back-dating. Imagine Internet distribution of your software with easy e-commerce features and automated authorizations without needing specialized equipment!

www.crypkey.com

InterLok™

Copy Protection • Electronic Software Distribution • License Management

Features:

- *Wraps Your Software In Minutes*
- *Try-Before-You-Buy & Immediate Purchase*
- *Software Metering & Rentals*
- *Windows & Macintosh Compatibility*
- *Foreign Language Support*
- *Key Diskette Option*

Sales: (408) 297-7444 ext.1

www.paceap.com

© 1998 PACE Anti-Piracy. All rights reserved worldwide. InterLok is a trademark of PACE Anti-Piracy.

TOOLS

C, C++, and JAVA DOCUMENTATION (v. 8.0)

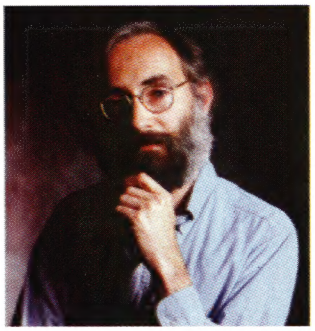
- **C-CALL ($69)** Graphic-tree of caller/called function hierarchy, cross-reference, file/function index.
- **C-CMT ($69)** Creates/inserts/updates comment-blocks (functions/identifiers used) for each function.
- **C-METRIC ($59)** Calculates path complexity, counts lines with comments, code, 'C' statements.
- **C-LIST ($69)** Lists and action-diagrams, or reformats source into user-selected standard formats.
- **C-REF ($69)** Creates cross-reference of local/global/define/parameter identifiers, class trees.
- **C-DOC ($199) PACKAGE** All 5 programs integrated as 1 overall C-DOC program. <10,000 lines. JavTREE graphic-tree viewer ($free in C-DOC).
- **C-DOC Professional ($299)** DOS, Win95/NT, OS/2, 1,000,000+ lines.
- **VERSION 8.0!**
  30-Day Money-back guarantee.

SOFTWARE BLACKSMITHS INC.
6064 St Ives Way, Mississauga
ONT Canada L5N-4M1
email @ swbs.com
Voice/Fax **(905) 858-4466**
http://www.swbs.com

Where do you find 116,000 professional software developers... monthly?

MARKETPLACE

LINUXWORLD
CONFERENCE & EXPO
BRIDGING THE [R]EVOLUTION OF THE LINUX SOLUTION
Jacob Javits Center
New York City
Conference: February 1- 4, 2000
Expo: February 2 - 4, 2000
Take control of your operating system, the power of innovation is in your hands! Linux users, softwaredevelopers, and businessleaders, participate in the evolution of the enterprise. LinuxWorld Conference & Expo in New York City is the premiere event, for supporters, investors and those interested in the linux operating system. Hosted in the financial capital of the world, the knowledge shared at this event will reinvent the way technology and business function in the new millennium.
For more information on exhibiting, please contact:
LinuxWorld Conference & Expo
3 Speen Street, Suite 320
Framingham, MA 01701
For more information or to register, please visit:
www.linuxworldexpo.com
or call 800.647.1474
Flagship Media Sponsor
Linux WORLD
Media Sponsors
LINUX MAGAZINE
UNIX/NT's PERFORMANCE COMPUTING
INFO WORLD
Sys Admin the journal for UNIX systems administrators
Tech Smart. Business Savvy.
NetworkWorld
Dr. Dobb's JOURNAL
Cornerstone Sponsor
VA LINUX Systems
Platinum Sponsors
LINUXCARE
SYBASE INFORMATION ANYWHERE.
IBM
IGEL
redhat
wned and Managed By
IDG
WORLD EXPO
www.linuxworldexpo.com



Out Time

Connect: Looking up host www.somewhereincyberspace.com...

"Counting the beats," the poet Robert Graves wrote:

Counting the slow heart beats,
The bleeding to death of time in slow heart beats...

"The bleeding to death of time." How much of my life, I wonder, is spilled as I sit staring through this seductive screen, waiting for a summoned web page to come to heel?

One could, perhaps,
Pen poems in the gaps.

Such web-wait verse would no doubt need to be of the personal, rather than the publishable, variety, but no less valuable for that.

Connect: Host www.somewhereincyberspace.com contacted. Waiting for reply...

Odd that we have chosen to use the word valuable where valued would seem better to fit the intended meaning. Maybe the choice means something about the meaning; maybe language is telling us that we need to take note of the fact that value is something we are free to bestow, rather than a fixed measure assigned by the market. Perhaps one could measure web delays in verse forms: That was just a haiku hiatus. Yesterday I had a two-sonnet wait connecting to dub-dub-dub dee-dee-jay dot com. That's nothing, this morning trying to get to dub-dub-dub linux dot org, I wrote an ancient Finnish epic. UNIX has been around so long that there is a rumor that Linus Torvalds didn't write Linux, he just translated it from the original runes. That's a myth, of course; everyone knows how Linux got started (but only Linus knows the Finnish).

In the 1578 Peter Borne Missza edition, Rune XX of the Finnish epic the *Kalevala* speaks of time and of another fascination of Linux users:

Time had gone but a little distance,
Scarce a moment had passed over,
Ere the heroes came in numbers,
To the foaming beer of northland...

Writer G. K. Chesterton claimed that poets do not go mad. "Mathematicians go mad...but creative artists very seldom." The source I am quoting from drops Chesterton at this point like an AOL connection, and I am left unenlightened as to whether Chesterton meant to imply that poetry kept one sane, or that poets are all crazy to begin with. From the poet's point of view, perhaps it's a distinction without a difference. Writer Stephen Harrod Buhner claims that in the craziness of alcoholic inebriation is the historic origin of poetry, so maybe that shines some light in the Chesterton haziness. Personally, I think any scholar whose definition of creative artist excludes mathematicians is a little crazy.

Connect: Host www.somewhereincyberspace.com contacted. Waiting for reply...

The window of my office overlooks the lawn of the restaurant of which I am part owner. As I wait for my web page to download, I watch the scene below. A customer fidgets, tugging at garments seen and unseen, as she sits in a chair under an umbrella while an apron-wearing employee bustles about in the sun filling orders. I, like the customer, sit waiting for my order to be filled. Of the three of us, the only one clearly not waiting for anything is the one called a waiter. The waiter, I muse, is really a server, and the waiting customer a browser. Or will be when she quits waiting because she has been served something to browse on. I sit at my desk fidgeting like a kid in a doctor's waiting room, musing that "patient" is an amazingly inappropriate word for such sufferers.

Patience, *n.* A minor form of despair, disguised as a virtue.
—Ambrose Bierce.

If you're a writer, you can only muse so long before the muse grabs you by the typing fingers; so I soon find myself versifying:

There was an old man who wrote COBOL
And gave the cold shoulder to SNOBOL,
But about Y2K
He had just this to say:
In my day Running Light was thought noble.

Connect: You have been disconnected.

Michael Swaine
editor-at-large
mswaine@swaine.com

To Get to Market Faster

Get a Firm Grip on Your e-Security.

RSA BSAFE® security components give you the control you need to beat the competition.

Developers, start your applications. The race to e-business is on. Whether you're developing apps for web business, general client-server, remote management or cell phones, one thing is clear: You need to get a firm grip on e-security or face a big potential pile-up.

Never fear, the RSA BSAFE pit crew is here. To maximize your opportunities for success in e-business, you need maximum security from the industry leader, RSA Security. Our RSA BSAFE security components give you all the tools you need to rev up SSL in C, SSL in Java™ and S/MIME implementation way beyond what street-issue browsers can offer. Such as full strength crypto available worldwide,* more extensive sample code and protocol links for services such as Telnet, FTP, NNTP, SMTP and POP3.

By combining SSL and S/MIME security protocols with the industry's most trusted cryptography, RSA BSAFE components give you the ability to negotiate the twists and turns of security protocol implementation in record time. And come out safely in the lead.

All our tools build confidence. Maneuverability and creative thinking will help you win the race, and with RSA BSAFE components you have the power to choose how our air-tight e-security works best within your application. You control the interface, error recovery and the best balance of strength vs. accessibility for your needs.

Best of all, you run with the confidence that your e-security operates independently of changes to the browser. No yellow caution flags for separate browser versions, either. With RSA BSAFE components, your apps will be completely interoperable.

Welcome to the winner's circle. It takes a winning team to win the race. RSA Data Security has been a driving force behind putting SSL and S/MIME on the map as industry standard security protocols. Today we're known as RSA Security, and no one knows more about how to build secure applications. So you can count on us to provide tools you can use to meet your goals on time and on budget.

It's never too late to get a jump start on the competition – especially when the stakes are so high. So don't waste another minute idling. **Qualify today for a free RSA BSAFE SSL-C, SSL-J or S/MIME-C software development kit* and head to the starting line.**

1.800.782.5453 or www.rsasecurity.com/win

RSA SECURITY™

The Most Trusted Name in e-Security

*Export or other restrictions may apply. RSA, BSAFE and the RSA Security logo are trademarks or registered trademarks of RSA Security Inc. Java is a trademark of Sun Microsystems, Inc. © 1999 RSA Security Inc. All rights reserved.

One of these four developers absolutely loves their Java™ IDE.

The rest have discovered the limitations of other developer tools. With VisualCafé Database Edition, you've got all you need to get your job done on time and, who knows, maybe have some time leftover for you. Whether you're a seasoned Java programmer, or moving to Java from Visual Basic® or C++, you'll find VisualCafé easy to learn and use because of its visual development environment—complete with automated wizards and code helper. In fact, if you're creating **servlets**, **JavaBeans**,™ or deploying comprehensive database-driven applications with **JDBC support**, it's your best choice for development.

VisualCafé
Database Edition

It offers superior speed, robust code, drag-and-drop programming, **QBE** (Query-By-Example) capability, **open Java API**, and support for data binding **JFC/Swing**. Call or visit our website and request our Java hierarchy poster. Join the tens of thousands of developers who use VisualCafé. You too may develop an attachment to it.

1-888-822-3409
www.visualcafe.com/vc3/javaposter

Symantec and the Symantec logo are U.S. registered trademarks of Symantec Corporation. VisualCafé is a trademark of Symantec Corporation. Java, JavaBeans and all Java based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the U.S. and other countries. Visual Basic is a registered trademark of Microsoft Corporation, in the U.S. and other countries. Other brands and products are trademarks of their respective holder/s.
© 1999 Symantec Corporation. All Rights Reserved.

The #1-selling Java IDE for 3 years running.

SYMANTEC